



LAUREA
AMMATTIKORKEAKOULU
Yhdessä enemmän

Työasemien hallinta pilvipalveluista tuotettuna CASE: Microsoft Intune

Savolainen, Sami

2017 Laurea

Laurea-ammattikorkeakoulu

Työasemien hallinta pilvipalveluista tuotettuna
CASE: Microsoft Intune

Savolainen, Sami
Tietojärjestelmäosaaminen
Opinnäytetyö
Marraskuu, 2017

Sisällys

1	Johdanto.....	7
1.1	Aihepiiri ja työn tausta.....	7
1.2	Tutkimuksen tavoitteet ja rajaukset	8
1.3	Tutkijan tausta	8
1.4	Tutkimuksen hyödynnettävyys	8
2	Metodologia	9
2.1	Tapaustutkimus.....	9
2.1.1	Suunnittelu	10
2.1.2	Valmistelu	11
2.1.3	Tiedon kerääminen.....	11
2.1.4	Kirjalliset lähteet	12
2.1.5	Osallistuva havainnointi.....	12
2.2	Suunnittelutieteellinen tutkimus	12
2.2.1	Suunnittelututkimuksen käsittelymalli	14
2.3	Tutkimusprosessin kulku	15
2.4	Aineiston kerääminen	16
2.4.1	Kirjallinen aineisto	16
2.4.2	Haastattelut	17
2.4.3	Havainnointi	17
2.5	Analysointi.....	18
2.6	Tutkimusattribuutit ja analysointiyksikkö	19
3	Teoreettinen viitekehys.....	20
3.1	Pilvilaskenta ja pilvipalvelut	20
3.1.1	Perusteknologiat.....	21
3.1.2	Pilvipalvelumallit.....	22
3.1.3	Käyttömallit	25
3.2	Multitenanttisuus.....	29
3.2.1	Määritelmiä.....	29
3.2.2	Miten multitenanttisuutta voidaan lähestyä?	30
3.2.3	Jakaminen vs. eristäminen	31
3.2.4	Multitenanttisuus infrastruktuuritasolla.....	32
3.3	Multitenanttisuus sovellustasolla	33
3.3.1	Sovellusarkkitehtuuri	34
3.3.2	Pilvisovelluksen valmiusasteet multitenanttiin käyttöön	35
3.3.3	Metatieto	36
3.3.4	Todennus ja valtuutus.....	36
3.3.5	Tietomalli.....	37

3.4	Yritysmobiilihallinta	39
3.5	Työasemien elinkaarenhallinta	40
3.6	System Center Configuration Manager	42
3.6.1	Esiasennusominaisuudet.....	42
3.6.2	Sovellushallintaominaisuudet	43
3.6.3	Päivitysten hallintaominaisuudet.....	45
3.6.4	Raportointiominaisuudet.....	45
3.6.5	Asetusten hallinta.....	46
3.6.6	Etähallintaominaisuudet	47
3.7	Microsoft Intune	48
3.7.1	Yleisesti	48
3.7.2	Työasemalaitteiden hallinta ja Windows 10.....	49
3.7.3	Hallintaominaisuudet Intune Clientilla ja MDM-hallinnalla	50
4	Tutkimustulokset.....	51
4.1	Vertailu hallintaominaisuuksista.....	51
4.2	Tapaustutkimuksen pohjalta havaitut kehityskohteet.....	53
4.3	Provisiointisovellus.....	54
4.3.1	Sovelluksen kehitysalusta	55
4.3.2	Provisiointisovelluksen paketointi ja julkaisu Intunessa	56
4.3.3	Suunniteltu työaseman käyttöönottoprosessi	57
5	Keskustelu ja arviointi	59
5.1	Tapaustutkimuksen arviointi	60
5.2	Suunnittelutieteellisen tutkimuksen arviointi.....	61
	Kuvat	68
	Taulukot	69

Sami Savolainen

Työasemien hallinta pilvipalveluista tuotettuna CASE: Microsoft Intune

Vuosi	2017	Sivumäärä	69
-------	------	-----------	----

Nykyisin yritysten ja yhteisöjen toimintaympäristö ulottuu suurelta osin perinteisen fyysisen toimitilan ulkopuolelle ja lisääntyvässä määrin kaikki työ tapahtuu muualla kuin tietyssä kiinteässä osoitteessa määrättyä aikana. Tämänkaltaisen kehitys on pakottanut yritykset miettimään uusia tapoja päätelaitehallinnan toteuttamiseen ja nykyaikainen ratkaisu on ostaa ja tuottaa palvelut pilvestä. Mobiililaitteiden hallinnan osalta tämä on itsestään selvää, varsinaisten Windows-työasemien osalta hallintaominaisuudet ovat vielä kohtuullisen lapsenkengissä. Sisäverkossa käytettävien palveluiden siirtäminen ja muuntaminen pilveen sekä samalla uusien toimintatapojen opettelu näyttäisi olevan eriasteisesti haastavaa yrityksissä.

Tässä opinnäytetyössä tutkittiin tapaustutkimuksena yleisesti työasemien hallittavia ominaisuuksia ja selvitettiin, miten näitä voidaan hallita sisäverkossa Microsoftin System Center Configuration Manager -tuotteella ja vastaavasti pilvipalvelussa Microsoft Intune -tuotteella. Lopputuloksena syntyi seitsemän vertailutaulukkoa, joiden pohjalta voidaan arvioida asiakasyrityksen kypsyyttä siirtyä pilvipalveluiden käyttöön myös työasemahallinnan osalta. Avoimen haastattelun pohjalta asiakaspalavereissa ilmeni yhtenä suurimpana kynnyskysymyksenä esiasennusmahdollisuuden puute pilvipalveluissa.

Tutkimuksen toisessa osassa suunniteltiin suunnittelutieteellisenä tutkimuksena käyttöönottoeli provisiointisovellus, jonka avulla voidaan merkittävästä madaltaa kynnystä itsepalvelukseen ja täysin pilvipohjaiseen työaseman käyttöönottoon paikasta ja ajasta riippumatta. Kyseinen ratkaisu tuo esiasennusta vastaavan toiminnallisuuden saataville pilvestä kevyesti ja kustannustehokkaasti; yrityksen ei tarvitse käyttää välissä esiasennusta suorittavaa tahoa vaan kaikki tarvittavat sovellukset ja asetukset tuotetaan pilvipalvelusta suoraan käyttäjälle.

Asiasanat: suunnittelutiede, tapaustutkimus, Microsoft, Intune, Azure, System Center, Configuration Manager, SCCM, Sovellus, Windows 10, itsepalvelu, pilvi, pilvipohjainen, moderni hallinta

Sami Savolainen

Workstation management as a cloud-based service CASE: Microsoft Intune

Year	2017	Pages	69
------	------	-------	----

Today, the operational environment for businesses and communities extends largely outside the traditional physical office and work is done increasingly somewhere else instead of a fixed location. This kind of evolution has forced companies to think about new ways to implement terminal device management and the modern solution is to buy and utilize cloud services. Regarding mobile device management, this is obvious, but with the actual Windows workstations, the management features are still evolving. Transferring and converting on premise services to the cloud, as well as learning new ways of working seems to be challenging for companies on different levels.

In this thesis, a case study was carried out to investigate the management features of Windows workstations and to find out how they can be managed on premise by using Microsoft System Center Configuration Manager and similarly in cloud service with the Microsoft Intune management product. As a result, seven comparison charts were formed, based on which the client company's maturity for switching to cloud services also for workstation management could be evaluated. Based on an open interview in customer meetings, one of the biggest issues was the lack of pre-installation opportunity in cloud services.

In the second part of the study, a provisioning application was planned as a design science research, which can significantly lower the threshold for cloud-based self-service workstation deployment, regardless of location and time. This solution makes the pre-installation functionality available from the cloud easily and cost-effectively. Clients don't have to use the intermediary pre-installer company, but all the necessary applications and settings are produced by the cloud service directly to the user.

Keywords: research science, case study, Microsoft, Intune, Azure, System Center, Configuration Manager, SCCM, Application, Windows 10, self-service, cloud, cloud-based, modern management

1 Johdanto

1.1 Aihepiiri ja työn tausta

Nykyisin yritysten ja yhteisöjen toimintaympäristö ulottuu suurelta osin perinteisen sisäverkon ja fyysisen toimitilan ulkopuolelle. Kasvavassa määrin jopa kaikki työ tapahtuu muualla kuin kiinteässä osoitteessa. Tietohallinnoille tämä aiheuttaa niin tietoturvan kuin käytännön elinkaaren hallinnan osalta suuria haasteita, koska yleisesti hallintatyökalut ja -käytännöt perustuvat siihen oletukseen, että esimerkiksi fyysisiä päätelaitteita käytetään maantieteellisesti suppealla alueella tai ainakin niiden oletetaan säännöllisesti vierailevan tietyissä toimipisteissä. Varsinkin pienillä ja keskisuurilla yrityksillä on haasteena pitää sopivaa määrää IT-henkilöstä varustettuna riittävällä osaamisella kustannustehokkaasti päivittäistoiminnan turvaamiseksi.

Näiden haasteiden pohjalta yritykset ovat nykyisin halukkaita siirtämään palveluita pilvestä tarjottaviin hallintapalveluihin, mikä tuo kustannustehokkuutta, kun pystytään purkamaan omissa tai perinteisen palveluntarjoajan tiloissa olevia palveluita pois ja siirtämään osittain käytön mukaiseen kustannusrakenteeseen. Nuorilla yrityksillä tämä on luonnollinen valinta, mutta vanhojen yritysten siirtyminen on aina haasteellisempaa niin käytännöllisistä kuin yrityksen kulttuuriinkin liittyvistä syistä.

Päätelaitehallinnan osalta haasteena on niin uusissa kuin vanhoissakin yritysympäristöissä vaihtelevasti vakioitu päätelaite- ja sovellusvalikoima, mikä aiheuttaa haasteita ja lisäkustannuksia elinkaaren hallintaan. Hallintatyövälineiden siirtyessä vähitellen multitenantteihin, vakioituihin pilvipohjaisiin palveluihin, on ajankohtaista miettiä, miten perinteiset sisäverkoissa olevat päätelaitehallintajärjestelmät voitaisiin korvata SaaS (software as a service)-pohjaisella palvelualustalla.

Silloin, kun toimintaympäristössä tapahtuu muutoksia esimerkiksi ICT-infran hallinnan osalta, tulee väistämättä eteen myös tarkasteltavaksi totutut toimintamallit ja niiden sopeuttaminen uuteen ympäristöön. Tämän hetken haasteina pilvipohjaisessa päätelaitehallinnassa on työasemamaailman osalta esiasennuksen puuttuminen, mikä koetaan edelleen tärkeäksi osaksi elinkaaren hallintaa.

1.2 Tutkimuksen tavoitteet ja rajaukset

Tutkimuksen ensimmäisenä tavoitteena on selvittää tapaustutkimuksen menetelmillä vertailtavia hallintaominaisuuksia multitenantin pilvipohjaisen Microsoft Intune -hallintapalvelun ja asiakasympäristökohtaisen System Center Configuration Manager -hallintatuotteen välillä ja näin nostaa esille palvelussa olevia oleellisia tekijöitä, jotka voivat hidastaa tai estää yrityksiä korvaamasta nykyistä sisäverkossa olevaa hallintajärjestelmää.

Tutkimuksen toinen tavoite on tuottaa lisäpalveluratkaisu suunnittelutieteellisenä tutkimuksena yhden havaitun puutteen ratkaisemiseksi. Tämä ratkaisun lisätavoitteena on tuotteistaa ja kaupallistaa myöhemmin tutkijan työnantajalle.

Tutkimuksessa keskitytään elinkaarihallinnan (hankinta, käyttöönotto, hallinta, poisto) piiriin kuuluviin käyttöönotto- ja hallintaosioihin ja elinkaaren laidoilla olevat hankinta- ja poistosiot rajataan tutkimuksen ulkopuolelle.

1.3 Tutkijan tausta

Työskentelen johtavan suomalaisen konsultointiyrityksen palveluksessa, joka tuottaa yritysasiakkaille pääasiassa Microsoft Azure -pilvipalveluihin ja Microsoft System Center -tuoteperheeseen pohjautuvia moderneja ratkaisuja. Tutkijan oma työnkuva liittyy asiakasympäristöjen päätelaitehallintaan Microsoft System Center Configuration Manager- ja Microsoft Intune -hallintatuotteilla. Lisäksi Windows 10 -työasemavakiointi on keskeinen osa aihepiiriin liittyvää tehtäväkenttää.

Opintojeni aikana aloitin työskentelyn nykyisen työnantajani palveluksessa ja syvennyin nopeasti pilvipalveluiden maailman oman erikoisalueeni, päätelaitehallinnan, puitteissa ja käytännössä kyseessä on Microsoft Intune -hallintajärjestelmä. Nopeasti oli havaittavissa se, että tuotteesta puuttuu helposti tuotettava käyttöönottopaketti ja heräsi ajatus tällaisen kehittämisestä. Olen havainnut, että asiakkailla on selkeästi kynnys ottaa Intunea työasemahallintaan esimerkiksi juuri perinteisen esiasennus-ominaisuuden puuttumisesta johtuen.

1.4 Tutkimuksen hyödynnettävyys

Tutkimus on ajankohtainen päätelaitehallinnan alueella nyt, kun siirtyminen pilvipalveluiden hyödyntämiseen kiihtyy kasvavalla vauhdilla. Päätelaitehallinnan osalta Microsoft Intune kehittyy myös, mutta tällä hetkellä asiakasympäristöissä hankitun kokemuksen perusteella on havaittu puutteita, jotka hidastavat päätelaitehallinnan siirtämistä kokonaan pilvipalvelupohjaiseksi. Tapaustutkimuksen tuloksena tuotetut vertailutaulukoita voidaan käyttää pää-

töksenteon pohjana valittaessa sisäverkkotuotteen ja pilvipalvelutuotteen välillä. Suunnittelutieteellisen tutkimuksen lopputuloksena syntynyt provisiointityökalu ratkaisee yhden oleellisen hidasteen pilvipalvelupohjaisen hallintatuotteen valitsemiseksi.

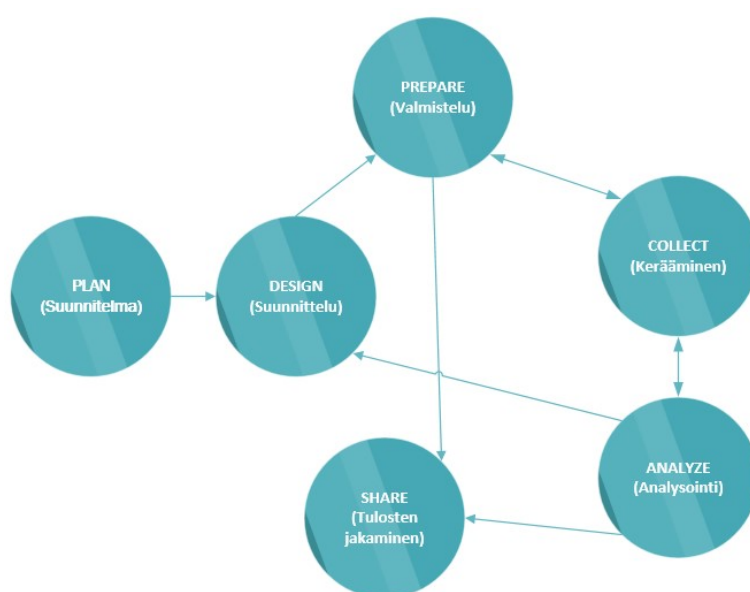
2 Metodologia

2.1 Tapaustutkimus

Tapaustutkimus (engl. case study) on tutkimusstrategia, joka soveltuu käytettäväksi silloin, kun tutkimuksen tavoitteena on ymmärtää ilmiötä, rakentaa uutta teoriaa tai testata sen paikkansa pitävyyttä esimerkkitapauksia tutkimalla. Yinin (Yin, 2014) mukaan tapaustutkimuksessa lähtökohtaisena tavoitteena on kerätä paljon monipuolista aineistoa tutkimuskohteen kartoittamiseksi. Tutkimustuloksilla ei kuitenkaan yleensä tuoteta yleistettävää uutta teoriaa tai mallia vaan pikemminkin lisätään ymmärrystä valitusta kohteesta tietystä näkökulmasta.

Tapaustutkimus voidaan tehdä yhdestä tai useammasta valitusta tapauksesta sekä lisäksi yksittäistä tapausta voidaan tutkia tarkemmin eri tasoilla tutkimustavoitteiden vaatimalla tarkkuudella. Yinin (2014) mukaan tapaustutkimus soveltuu käytettäväksi strategiaksi mm. silloin, kun halutaan vastauksia kysymyksiin ”miksi” tai ”miten”.

Yin kuvaa tapaustutkimusta iteratiivis-lineaariseksi prosessiksi (kuva 1), jossa edetään vaiheittain tavoitetta kohti, kuitenkin tarvittaessa palaten edellisiin vaiheisiin iteratiivisesti.



Kuva 1: Yinin (2014) lineaaris-iteratiivinen prosessi.

Tapaustutkimuksen tarkoituksena on luoda ymmärrettävä kuva valitusta tapauksesta, joka voi olla osa varsinaisen tutkimuksen esitutkimusta tai sitten tapausta voidaan tutkia puhtaasti sen itsensä vuoksi (Bamberg, Jokinen, & Laine, 2007). Sopivan tapauksen valinta ja edelleen tapaustyyppin hahmottaminen vaikuttavat muun muassa siihen, miten yleistettävää (Bamberg et al., 2007) tietoa tapauksen tutkimustuloksilla voidaan tuottaa.

2.1.1 Suunnittelu

Tapaustutkimuksen suunnitteluvaiheessa määritellään yksi tai useampi tutkimuskysymys, jotka vastaavat tyypillisesti kysymyksiin miten tai kuinka (Yin, 2014). Kysymyksen valinnan pohjalta tehdään yleensä kirjallisuustutkimusta, jonka pohjalta Yinin (Yin, 2014) mukaan voidaan päätyä valitsemaan liian helposti jo aiemmassa tutkimuksessa käsitelty tutkimuskysymys ja todellisuudessa saman tutkimuksen toistaminen ei välttämättä tuota uutta tietoa. Tällaisessa tilanteessa on hyödyllistä miettiä, voidaanko samasta viitekehystä valita toinen analysointiyksikkö, jossa keskitytään tuottamaan perustellusti tarkempaa tietoa kokonaisuuden ymmärtämiseksi.

Tutkimuskysymys itsessään ei yleensä suoraan kerro tutkijalle, mitä tutkimuskontekstissa todellisuudessa tulee tutkia, jotta varsinaisiin asetettuihin pääkysymyksiin voidaan esittää luotettavan tutkimusdatan avulla (Yin, 2014). Oleellisin osa suunnitteluprosessia onkin tutkimusyksikön valitseminen eli varsinainen tutkittava tarkasti rajattu kokonaisuus, joka Yinin (Yin, 2014) mukaan jaetaan vähintään kahteen vaiheeseen, joista tapauksen määrittelyvaiheessa valitaan yksi tai useampia tutkimuskohteita, esimerkiksi haastateltavia henkilöitä. Tutkimuksen edetessä tehdään tarkempi rajaaminen, esimerkiksi haastateltujen henkilöiden valitsemista sopivaksi tutkimusaineistoksi ja tässä vaiheessa voidaan iteratiivisesti palata tarkentamaan tutkimuskontekstia ja tarkemmin valitsemaan sopivia analysointiyksiköitä eli haastateltavia.

Tutkimuskysymystä ja analysointiyksikköä valitessaan tutkijan on kysyttävä itseltään, mitä tapauksesta voidaan oppia ja miten tutkimuskohteesta saadaan lisää oleellista uutta tietoa, jotta ymmärrystä tutkimuskohteesta voidaan parantaa (Bamberg et al., 2007). Tutkija ei välttämättä tiedä tutkimuksen alussa tarkkaan, mikä on täsmällinen analysointiyksikkö vaan lähtökohtana on pikemminkin tunne (Bamberg et al., 2007) kiinnostavaan ja ajankohtaiseen aiheeseen, joka tarkentuu ja jäsentyy tutkimusprosessin kuluessa.

Tapaustutkijan tulee ymmärtää käsitteiden tapaus ja tutkimuskohde ero, joista tapaus on itse asiassa analysointiyksikkö, mikä omalta osaltaan realisoi tutkimuksen kohteena olevaa tutkimusobjektia (Bamberg et al., 2007). Mikäli tutkimustiedon keräämisvaiheessa on löydetty useita potentiaalisia analysointiyksiköistä, joista analyysivaiheessa ei pystytäkään eristämään

yhtä tai mahdollisesti muutamaa tutkimuskysymykseen soveltuvaa analysointiyksikköä, on mahdollisesti kysymys siitä, että tutkimuskysymyksen asettelu ei ole tarpeeksi rajattu tai on muuten jäänyt liian epämääräiseksi (Yin, 2014).

2.1.2 Valmistelu

Tapaustutkimuksessa suunnitteluvaiheeseen kuuluu tutkimuskysymyksen suunnittelu, kun taas valmisteluvaiheessa aloitetaan lähdetiedon hankkiminen ja tutkimustiedon kohteiden etsiminen (Yin, 2014), joiden pohjalta voidaan tarkentaa tutkimuskysymyksen asetantaa ja hahmottaa mahdollisia analysointiyksiköitä. Tutkijan tulee olla mukautuva (Yin, 2014) tässä vaiheessa, jotta tutkimuskysymyksen ympärille hankittava tutkimustieto on tarkoituksenmukaista ja tarvittaessa palattava iteratiivisella otteella miettimään, voidaanko tutkimuskysymystä tai analysointiyksikköä muuttaa ja tarkentaa.

Teoriapainotteisessa tapaustutkimuksessa tutkija kerää ensisijaisesti teoreettista tietoa (Bamberg et al., 2007), mistä valitaan tietty tapaus eli analysointiyksikkö, jonka pohjalta testataan teorian paikkansapitävyyttä (Yin, 2014) ja pyritään tätä kautta rakentamaan tutkimuskohteen ympärille teoreettista viitekehystä. Tapaus voidaan määritellä myös käytännönläheisesti (Bamberg et al., 2007), jolloin vasta tutkimuksen edetessä voidaan muodostaa käsitys viitekehuksesta ja varsinaisesta tutkimuskohteesta. Varsinkin aineistotutkimuksessa ei välttämättä ole valmista teoriaa (Bamberg et al., 2007) vaan itse asiassa tutkija muodostaa teoreettisen viitekehysten tutkimuksen edetessä ja tarkentaa tutkimuksen analysointiyksiköitä ikään kuin valmiin teorian ja tutkimuksessa löydetyn tiedon vuoropuheluna.

2.1.3 Tiedon kerääminen

Tiedonkeruussa on mahdollista käyttää lukemattoman paljon erilaisia lähteitä, joita voi olla perinteisten haastatteluiden ja kirjallisuuden lisäksi liikkuvan kuvan, äänitysten ja muun sellaisen multimedian muodossa. Yin (Yin, 2014) tuo esille kuusi yleisintä lähdetyyppiä, joita ovat kirjalliset lähteet, arkistomateriaali, haastattelut, havainnointi ja seuraaminen, osallistuva havainnointi sekä fyysiset artefaktit, joista yksikään ei ole muita parempi sellaisenaan vaan tutkimuskohde ratkaisee soveltuvimmat lähdemateriaalit ja eri luokkiin kuuluvat lähteet voivat olla toisiaan täydentäviä (Yin, 2014).

Tämän tutkimuksen lähdemateriaalina käytettiin teoriapohjan luomiseen painettua ja sähköistä kirjallisuutta, jotka käsittelevät pilvipalveluihin liittyvää viitekehystä eri näkökulmasta ja tutkimuksen aikana tämän pohjalta muodostettiin multitenanttisuuden ymmärtämistä pilvipalveluiden eri tasoilla. Lisäksi taustalla on tutkijan oma lähes 20 vuoden työkokemus ICT-infrastruktuurialalla sekä tietohallinnon että palveluntarjoajan toimintasektoreilla, joista jäl-

kimmäisessä on moniasiakkuusympäristöissä noussut esille tarve palveluiden vakioinnista ja tarjoamisesta yhteisten ja keskitettyjen palveluiden muodossa kaikille asiakkuuksille sen sijaan, että kullekin asiakkuudelle perustetaan oma erillinen palvelinympäristö eli instanssi.

2.1.4 Kirjalliset lähteet

Kirjallisilla lähteillä tarkoitetaan esimerkiksi perinteistä painettua tai vastaavaa sähköisessä muodossa julkaistua kirjallisuutta, minkä etuna on kiinteä muuttumaton sisältö, johon voidaan palata tarvittaessa (Yin, 2014). Muita tähän luokkaan kuuluvaa materiaalia voivat olla julkaistut tutkimukset, artikkelit, sähköpostit, dokumentit, pöytäkirjat yms. kirjallinen materiaali. Suurin osa tähän luokkaan kuuluvasta materiaalista on saatavilla sähköisesti, jolloin siihen pääsy on paikasta riippumatonta, viiveetöntä ja kustannustehokasta. Huonona puolena muissa kuin varsinaisiksi painetuiksi tai sähköisiksi kirjoiksi luokiteltavissa materiaaleissa voidaan pitää niiden luotettavuutta ja sisällön tuottajien jäljitettävyyttä. Näitä seikkoja on syytä tarkastella kriittisesti pohdittaessa tutkimuksen validiteettia ja reliabiliteettia.

2.1.5 Osallistuva havainnointi

Osallistuva havainnointi tekee tiedonkeruumuotona tutkijasta aktiivisen toimijan, joka on läsnä toimintaympäristössä (Yin, 2014) ja jossain roolissa tutkittavan ilmiön ympärillä. Tutkija on tässä tapauksessa toiminut pitkään eri toimintaympäristöissä, joita on myöhemmin alettu kutsuaan pilvipalveluiksi ja käsite multitenantisuus on tullut ajankohtaiseksi, kun on alettu tiedostamaan tarve operatiivisen toiminnan tehostamisesta ja inhimillisten virheiden määrän vähentämisestä luomalla vakioituja ja keskitettyjä palveluympäristöjä moniasiakkuusnäkökulma huomioiden.

2.2 Suunnittelutieteellinen tutkimus

Suunnittelututkimuksessa tavoitteena on rakentaa ja kehittää artefakteja eli tuotoksia, joilla ratkaistaan reaalielämässä esiin tulleita ongelmia esimerkiksi yritysorganisaation toimintaympäristössä (Peffer, Tuunanen, Rothenberger, & Chatterjee, 2008). Suunnittelututkimuksen tavoitteena on pitkäjänteisesti kehittää yleispäteviä menetelmiä, jotta tietojärjestelmien suunnittelua ei tarvitsisi aloittaa aina tyhjästä vaan hyödyntää jo aiemmin hankittua tietämystä. Sen lisäksi, että kehitetään uusia tuotteita, voivat artefaktit olla olemassa olevien ratkaisujen parannettuja ja jatkokehitettyjä versioita, joilla edelleen tehostetaan yrityksen tai yhteisön toimintaa (Hevner & Chatterjee, 2010). Lisäksi toimintaympäristöt ovat jatkuvassa muutoksessa ja tästä seuraa luonnollisesti pysyvä vaatimus tietojärjestelmien jatkuvasta kehitystarpeesta toiminnan tukemiseksi.

Suunnittelumenetelmiä hyödynnetään yleisesti lähes kaikilla tutkimusaloilla, ja tietojärjestelmäsuunnittelussa on alusta lähtien käytetty samoja suunnitteluperiaatteita kuin muillakin aloilla. Tietojärjestelmät ovat jatkuvasti muuttuvia laitteista, sovelluksista ja käyttöliittymistä muodostuvia kokonaisuuksia ja tästä erityispiirteestä johtuen asettaa suunnittelulle haasteita ja tarpeita uusille ideoille (Hevner & Chatterjee, 2010). Haasteista merkittävässä roolissa on ihminen eli käyttäjät, jotka loppujen lopuksi käyttävät tietojärjestelmiä. Tästä johtuen tietojärjestelmien tutkimuksessa käytetään suunnittelutieteen apuna käyttäytymistieteellistä tutkimusta, joka kehittää teorioita ja sitä kautta lisää ymmärrystä ihmisen ja ryhmien toiminnasta tietojärjestelmien parissa (Hevner, March, Park, & Ram, 2004).

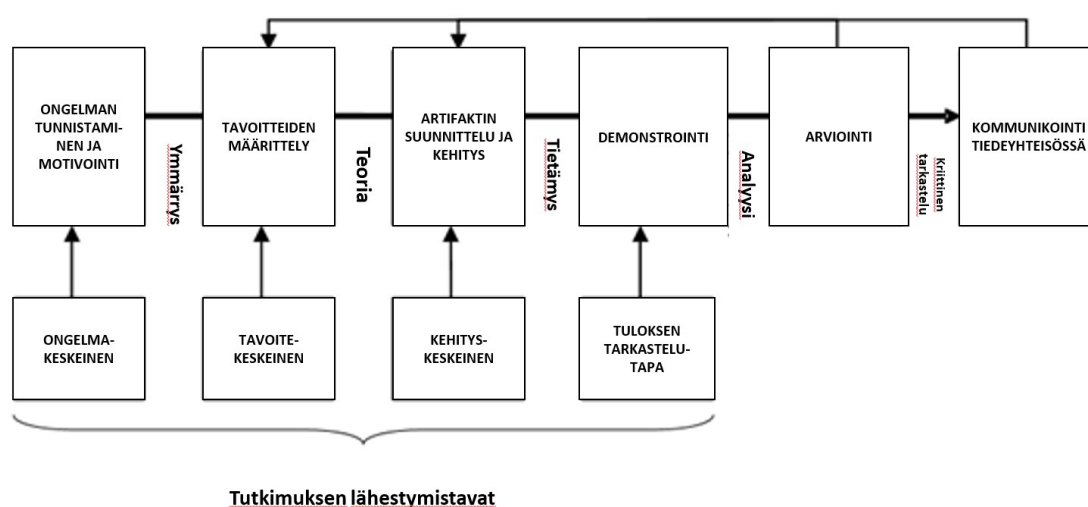
Suunnittelututkimus on tutkimusalana kiinnostunut keskeisesti artefaktista itsessään ottamatta erityisesti huomioon inhimillisiä tekijöitä ja ihmisiä, joita ei voida ei kuitenkaan voida erottaa kokonaisuudesta (March & Smith, 1995). Vaikka tietojärjestelmä onkin itsenäinen kokonaisuus, ei sitä voida kokonaan erottaa käyttäjästä ja näin syntyy keskinäinen vuorovaikutus käyttäjän ja järjestelmän välille. Hevnerin et al. (2004) mukaan nämä tieteen alat eivät suinkaan kilpaile keskenään vaan täydentävät toisiaan lähestyttäessä kokonaisuuden ymmärrystä ja kehitystä.

Tulkinnat suunnittelututkimuksen päämääristä vaihtelevat eri tutkijoiden tulkintojen mukaan. Hevner et al. (2004) mukaan suunnittelututkimuksessa luodaan ideoita ja tutkitaan mahdollisuuksia tuotteen kehittämiseksi, kun taas Nunamakerin ja Briggsin (2011) ajatuksena on luoda konkreettinen artefakti, reaalin lopputulos suunnittelututkimuksen päämääränä. Kumpaa-kin lähestymistapaa voitaneen pitää toisiaan täydentävänä, jolloin kokonaisuutena on selvitetty toimintaympäristöä ja mahdollisuuksia (Hevner et al., 2004) havaitun reaali maailman ongelman ratkaisemiseksi todellisen (Nunamaker & Briggs, 2011) artefaktin tuottamassa muodossa. Suunnittelututkimuksen päämääränä on lähestymistavasta riippuen luoda kehikoita tietojärjestelmäkehityksen tueksi, jolloin saadaan luotua yhteneväisiä käytäntöjä työprosessien tehostamiseksi. Haasteena on kuitenkin se, että toimintaympäristö määrittää pitkälle projektin sisällön ja kulun, eikä valmista mallia voine koskaan soveltaa suoraan reaalielämän kaikissa tilanteissa.

Reaali maailmassa törmätään erilaisissa toimintaympäristöissä tietojärjestelmäkokonaisuuksiin, joiden suunnittelussa ja toteutuksessa ei ole huomioitu lainkaan tai huonosti loppukäyttäjät eli ihmiset. Suunnittelutiede on yksi tutkimusaloista, jotka pyrkivät muun muassa etsimään syitä tietojärjestelmäsuunnittelun onnistuneisiin ja epäonnistuneisiin toteutuksiin (March & Smith, 1995). Suunnittelututkimuksen tarkoituksena on näin saada tarpeet ja toteutus kohtaamaan paremmin toisensa.

2.2.1 Suunnittelututkimuksen käsittelymalli

Suunnittelutiede luo malleja tietojärjestelmien ymmärtämiseen, tutkimukseen ja organisaatioissa havaittujen ongelmien ratkaisemiseksi (Hevner et al., 2004). Suunnittelun tuloksena syntyvät artefaktit voivat olla uusi tai parannettu käytäntö, malleja tai uusia tuotteita. Peffers et al. (2008) esittävät kuusivaiheisen suunnittelutieteellisen menetelmämallin, jossa tutkitaan tietojärjestelmiä ja esitetään omaan kokemukseen ja aiempaan tutkimukseen perustuvia parannusehdotuksia. Mallin (kuva 2) mukaan tutkimuksen kulku etenee johdonmukaisesti vaiheittain, mutta prosessia ei välttämättä tarvitse aloittaa prosessikaavion alusta.



Kuva 2: Suunnittelututkimuksen prosessimalli Peffersin (2008) mukaan.

Määritellään tutkimusongelma ja siihen vaikuttavat tekijät sekä lisäksi perustellaan artefaktin luomisen arvo (Peffers et al., 2008). Tutkimusongelman rajaamisen jälkeen selvitetään ratkaisuvaihtoehtoja ja tarvittaessa kokonaisuus voidaan jakaa ja mallintaa pienempiin osiin. Tutkimusongelman lähtökohtina voivat olla toiminallisessa ympäristössä tapahtuvat muutokset tai esimerkiksi paineet liiketoiminnan ja strategian tehostamiseen. Tutkimusongelman ymmärtäminen ja rajaaminen on tärkeää onnistuneen lopputuloksen näkökulmasta (van Aken & Romme, 2004).

Määritellään tutkimusongelman perusteella *tavoitteet*, jotka ratkaisulla tulee saavuttaa prosessin lopussa. Kyseessä voi olla uuden kehittäminen tai nykyisen parantaminen (Hevner & Chatterjee, 2010), mikä edellyttää nykytilanteeseen perehtymistä (Peffers et al., 2008). Valittavan ratkaisun tulee parantaa tai ratkaista havaittu ongelmatilanne ja suunnittelussa on huomioitava integroituvuus olemassa olevaan toimintaympäristöön kokonaisuuden toimivuuden vuoksi.

Suunnittelu- ja kehitysvaiheessa luodaan artefakti määriteltyjen tavoitteiden pohjalta (Hevner & Chatterjee, 2010), ja nyt vertaillaan saatavilla olevia vaihtoehtoja, joista valitaan sopivimmat tavoitteiden toteuttamiseksi. Prosessin tässä vaiheessa tutkijan oman ammattitaidon ja kokemuksen merkitys korostuu ja asiantuntemus lisääntyy tutkimuksen aikana tehtävien iteraatioiden kautta. Tässä vaiheessa voidaan myös soveltaa osallistavaa suunnitteluotetta (Saad-Sulonen, 2014) ja hyödyntää loppukäyttäjän (asiakas, yritys, henkilö) osaamista sovellettavalla osa-alueella.

Demonstrointivaiheeseen päästään prototyypin valmistuttua ja tämä vaihe toteutetaan esimerkiksi simuloimalla ratkaisua tai järjestämällä sopivanlaajuinen proof of concept -käyttöönotto tuotantoympäristössä, ja lopputuloksena saadaan ensimmäinen palaute toimivuudesta ja artefaktin soveltuvuudesta kohdeympäristöön (Hevner & Chatterjee, 2010; Peffers et al., 2008).

Arvioimalla demonstroinnissa saatavia tuloksia selvitetään artefaktin ja alun perin määritellyn ongelman vastaavuutta eli millainen lopputulos saavutetaan. Havaitut kehityskohteet voidaan palauttaa iteratiivisesti takaisin suunnittelu- ja kehitysvaiheeseen, jolloin parannetaan tarvittavaa tietämyspohjaa ja parannetaan artefaktin toimivuutta (Peffers et al., 2008).

Lopulta tutkimuksen ja projektin valmistuttua *tiedotetaan* asianmukaisesti tarvittavia osapuolia. Hevnerin (Hevner & Chatterjee, 2010) mukaan tieteellisessä tutkimuksessa tulee tutkimustulokset esittää siten, että ne ovat tiedeyhteisön saatavilla. Kuitenkin kysymyksessä voi olla yrityksen tuotekehitys, mikä rajaa liikesalaisuuksiksi luokiteltavien osien julkistamista.

2.3 Tutkimusprosessin kulku

Tutkimusongelman *määrittely* tapahtui ja muodostui vähitellen työskennellessäni erilaisissa asiakasprojekteissa ja kohdatessani käytännön tarpeita. Käytännössä tutkimus alkoi vähitellen ja opinnäytetyöaiheeksi Microsoft Intune aihepiirinä muodostui varsin myöhään. Tiedon kerääminen oli tässä vaiheessa jo suoritettu.

Opinnäytetyö muodostui *tavoitteiltaan* kaksiosaiseksi siten, että selvitetään tapaustutkimuksena Intunen nykyisiä ominaisuuksia ja mahdollisuuksia. Näistä poimittiin lopulta suurimmaksi esteeksi muodostuneen esiasennuksen puutteen korvaaminen jollakin uudella pilvipohjaisella ratkaisulla. Lopulliseksi tavoitteeksi muodostui itsepalvelullisen käyttöönottoratkaisun tuottaminen pilvestä työasemaympäristöön.

Suunnittelu- ja kehitysvaiheessa lisättiin tietämystä viitekehyksen aihepiirissä eli mobiililaittehallinnan, multitenanttisuuden ja Microsoft Intunen aihe-alueilla. Lisäksi Intunen osalta on

minulle kerääntynyt viimeisen vuoden aikana paljon syvällistä käytännön kokemusta erilaisten ympäristöjen ja käytännön haasteiden kautta. Suunnittelutieteellisessä tutkimuksessa lopputuloksen kannalta tutkijan kokemus ja asiaan perehtyneisyys ovat avainasemassa.

Tässä vaiheessa prosessia määriteltiin käyttöönottosovellukselta vaadittavat toiminnallisuudet ja näitä automatisoitiin aluksi käyttämällä perinteisiä komentorivitedostoja, joita edelleen paketoitiin MSI-paketeiksi jakelua varten. Tässä vaiheessa tuli useita puutteita ja virheitä esille liittyen Intunen ominaisuuksiin ja työasemapäässä käyttöoikeuksiin, joita jouduttiin ratkomaan iteroiden useita testauskierroksia. Lopulta löytyi myös kehittynyt Powershell-pohjainen avoimen lähdekoodin yhteistökehitetty tuote, jossa oli juuri tämänkaltaista tarkoitusta varten olemassa valmiita toimintoja ja näin päädyttiin kehittämään lopullinen ratkaisu kyseisen työvälineen päälle.

Demonstroitivaihe toteutettiin kaupallisella PoC (proof of concept) -projektina, jossa ratkaisusta kiinnostuneelle asiakkaalle suunniteltiin käyttöönottopaketti sisältäen halutut sovellukset ja asetukset. Pohjakehitys oli tehty jo valmiiksi siten, että ratkaisu soveltuu kaikille asiakkuuksille mahdollisimman pienillä muutoksilla ja lopulta kyseisen asiakkuuden määrittely ja toteutus oli suhteellisen nopeaa ja vaivatonta.

Ensimmäisen version testauksessa ja *arvioinnissa* ilmeni pieniä käytännön puutteita, jolloin palattiin iteratiivisesti edelliseen *suunnittelu- ja kehitysvaiheeseen* ja luotiin korjattu versio tuotteesta. Toisella demonstroitikierroksella toiminnallisuudet todettiin oikeanlaisiksi ja riittävän valmiiksi.

Tutkimuksen *tiedotusvaihe* tapahtui sovelletusti toimittamalla asiakkaalle dokumentaatio sovitusti tehdystä työstä ja käymällä palautekeskustelu projektin kulusta ja lopputuloksesta. Tältä pohjalta voidaan kehittää lisää tarvittavia toiminnollisuuksia jo tässä vaiheessa toimivaksi todettuun tuotteeseen.

2.4 Aineiston kerääminen

2.4.1 Kirjallinen aineisto

Aineiston kerääminen aloitettiin kirjalliseen aineistoon perehtymällä, mikä realisoitui Microsoftin viralliseen tuotedokumentaatioon (Microsoft Corporation, 2017b) sekä Microsoftin vastaavan tuotekehitystiimin blogikirjoituksiin (Microsoft Corporation, 2017a). Lisäksi käytettiin lähteinä tunnettujen MVP-tason asiantuntijoiden (Arvidmark, 2017; Parker, 2016) blogeja kokonaisuuden hahmottamiseksi ja ideoiden luomisen pohjana. Blogikirjoitusten käyttäminen edellyttää suurta kriittisyyttä tiedon todenmukaisuuden ja oleellisuuden kannalta, sillä esitet-

ty aineisto ei välttämättä ole objektiivista tutkimusongelman kannalta (Kananen, 2013). Kuitenkin on muistettava, että tutkijalla on aihepiirin alueelta pitkä työkokemus, mikä antaa hyvän varmuuden lähteiden luotettavuutta arvioitaessa.

2.4.2 Haastattelut

Haastattelut ovat yksi yleisimmistä tiedonkeruumenetelmistä tapaustutkimuksessa ja tavoitteena on saada uutta tutkimatonta tietoa tutkimuskohteesta (Järvinen & Järvinen, 2004). Haastattelut voidaan jakaa kolmeen kategoriaan: strukturoituihin, puolistrukturoituihin ja avoimiin riippuen siitä, millä tarkkuudella haastattelukysymykset on laadittu (Järvinen & Järvinen, 2004). Haastattelun käyttäminen tiedonkeruumenetelmänä on, jonka etuina voidaan pitää joustavuutta aineistoa kerätessä (Hirsjärvi, Remes, & Sajavaara, 2009) sekä kohdeorganisaatioiden haastateltavien mahdollisuutta kertoa omina sanoin heitä koskettavia (Robson, 2002) asioita.

Tapaustutkimuksen tiedonkeruu tehtiin avoimena haastatteluna, mikä toteutui käytännössä sähköpostin sekä Skype-kokouksissa hallintajärjestelmän käyttöönoton eri vaiheissa. Yhdeksässä asiakastapauksessa (n=9) kerättiin normaalisti pre-sales -vaiheessa ja projektin käynnistyttyä suunnitteluvaiheessa tietoa asiakkuuksien toimintaympäristöstä, elinkaarenhallintaprosessista ja tulevaisuuden tarpeista muun muassa päätelaitehallinnan osalta ja tältä pohjalta muodostui tarvittavia ominaisuuksia ja elementtejä, joista muodostettiin analyysiaineisto. Strukturoidun kysymyspatteriston määrittely ennalta ei vaikuttanut soveltuvalta tässä vaiheessa, kun opinnäytetyön kokonaisuus oli vasta muodostumassa ja lisäksi tutkimus on luonteeltaan laadullinen, kun taas strukturoitu haastattelu soveltuu paremmin määrällisen tutkimuksen tiedonkeruumenetelmäksi (Järvinen & Järvinen, 2004).

2.4.3 Havainnointi

Havainnointia käytetään tiedonkeruumenetelmänä sekä kvalitatiivisessa että kvantitatiivisessa tutkimuksessa silloin, kun kohteesta ei löydy aiempaa tutkimusdataa (Kananen, 2013). Hirsjärven et al. (2009) mukaan havainnointitekniikat voidaan jakaa ääripäinä osallistuvaan ja systemaattiseen, ja osallistumisen aste voi olla jotain tältä väliltä. Tutkijan rooli on osallistuvassa havainnoinnissa mukana prosessissa vaikuttaen siihen eri asteisesti, kun taas systemaattisessa havainnoinnissa rooli on ulkopuolinen ja passiivisempi (Hirsjärvi et al., 2009).

Tässä tutkimuksessa havainnointi tapahtui testaamalla omakohtaisesti erilaisia hallintaominaisuuksia haastatteluissa ilmenneiden tarpeiden pohjalta pilviympäristössä. Tavoitteena oli todeta toiminnollisuuksien olemassaolo siten kuin Microsoft esittää asioiden olevan. Käytännössä osa toiminneista on dokumentoitu vaikeaselkoisesti ja ainoa tapa selvittää toimivuus on

suorittaa empiiristä havainnointia valitussa ympäristössä tutkijan oman kokonaisymmärryksen lisäämiseksi.

2.5 Analysointi

Kirjallisen aineiston analysoinnin pohjana oli tutkijan oma vahva työkokemus ja näkemys aihepiiristä sekä teoriasisällöllisesti että käytännön toteutustasolla. Tämän valmiin tietopääoman lisäksi perehdyttiin kahdesta vertailtavasta tuotteesta julkaistuun kirjallisuuteen ja erityisesti alan johtavien konsulttien julkaisemiin (Bennett, Daalmans, & Martinez, 2016; Coupland, 2014) yhteisteoksiin mahdollisimman kattavan kokonaiskuvan saavuttamiseksi. Vertailtavista tuotteista Configuration Managerin osalta löytyy erittäin laajasti tunnettujen konsulttien kirjoittamia seikkaperäisiä teoksia, joita käytettiin ominaisuuksien ryhmittelyn (Robson, 2002) runkona.

Microsoft Intune -tuotteesta sen sijaan ei löydy ajankohtaista julkaistua kirjallisuutta vaan ainoa ajantasainen yhtenäinen materiaali on Microsoftin tuottamaa virallista tuotedokumentaatiota. Muiksi haasteiksi muodostuivat Intune-hallinnan siirtymisen osaksi Azure-hallintaa ja kuukausittain lisääntyvät uudet hallintaominaisuudet. Kokonaisuuden muodostamisen apuna käytettiin alan tunnettujen asiantuntijoiden säännöllisiä julkaisuja (Arvidmark, 2017; Daalmans, 2017; Larsen, 2017), joita tutkija itse myös käytännön työssä käyttää tutustuessaan uudistuviin ominaisuuksiin. Pelkkien blogikirjoitusten perusteella olisi erittäin vaikeaa rakentaa monipuolista kokonaiskuvaa ilman tutkijan omaa kokonaisvaltaista ammatillista työkokemusta ja tarjotun tiedon kyseenalaistamista.

Avoimen haastattelun pohjalta yhdistettynä tutkijan omaan kokemukseen aihepiiristä muodostui käsitys kerätyn tiedon ryhmittelystä teemoihin siten, että hallintaominaisuudet ryhmiteltiin aihe-alueittain omiin taulukoihin.

2.6 Tutkimusattribuutit ja analysointiyksikkö

Tutkimusattribuuteilla ilmaistaan kyseisen tutkimuksen tieteellistä täsmällisyyttä. Tässä tutkimuksessa on käytetty sekä tapaustutkimuksen että suunnittelutieteellisen tutkimuksen menetelmiä ja tutkimusattribuutit (Taulukko 1: TutkimusattribuutitTaulukko 1) on koottu soveltuvasti tähän pohjautuen.

Taulukko 1: Tutkimusattribuutit

Analysointiyksikkö	Työasemahallintaohjelmiston ominaisuus.
Analysointitapa	Työasemahallintaohjelmiston ominaisuuksien ryhmittely.
Analysointikirjallisuus	Robson, C. 2002, Kananen (2013).
Merkitys	<p><i>Tapaustutkimus</i> tuotti jäsennellyn vertailutaulukon uuden ja vanhan hallintatavan hallintaominaisuuksista päätöksen- teon pohjaksi suunniteltaessa käyttöönottoa yritysympäris- töissä.</p> <p><i>Suunnittelutieteellinen</i> tutkimus tuotti prototyypin yhden kokonaan puuttuvan ominaisuuden (laitteen esiasennus) täyttämiseksi itsepalvelullisesti. Tuotteen avulla voidaan madaltaa yritysten kynnystä siirtyä käyttämään puhtaasti pilvipohjaista työasemahallintaa.</p>
Metodologia	<p>Opinnäytetyön tapaustutkimuksen aineistoa kerättiin pereh- tymällä Microsoftin Intune-tuotteen dokumentaatioon, ai- hetta käsitteleviin blogeihin, asiakashaastatteluilla ja ha- vainnoimalla ominaisuuksia käytännön testeissä. Kerätty data ryhmiteltiin hallintaominaisuuksien mukaisesti vertai- lutaulukoihin, mistä saatiin tutkimuskysymykseen vastaus.</p> <p>Suunnittelututkimusprosessin malliksi valittiin Peffersin (Peffers et al., 2008) esittämä DSRP-menetelmä, jonka ai- neistona käytettiin tapaustutkimuksessa valmiiksi analysoi- tua dataa. Analyysin perusteella valittiin yksi puuttuva omi- naisuus, johon kehitettiin ratkaisuprototyyppi iteroiden eri vaiheita ja tuloksena saatiin artefaktina itsepalvelullinen käyttöönottosovellus.</p>
Metodologiakirjallisuus	<p>Tapaustutkimus: Yin 2009.</p> <p>Suunnittelututkimus: Hevner & Chatterjee 2010, Peffers et</p>

	al. 2008, van Aken & Romme 2004.
Otsikko	Työasemien elinkaaren hallinta pilvipalveluissa ja kehityskohteet CASE: Microsoft Intune
Teoriakirjallisuus	Microsoft Intune 2017, Arvidmark, 2017; Daahlmans, 2017; Larsen, 2017)
Tiedonkeruu	Kirjallinen aineisto: Tuotedokumentaatio, blogit. Asiakashaastattelut: avoin haastattelu. Havainnointi.
Tiedonkeruukirjallisuus	Hirsjärvi et. al (2009), Robson (2002), Järvinen & Järvinen (2004), Kananen (2013).
Tutkimuskohde	Työasemahallinnan mahdollisuudet pilvipalvelupohjaisesti ja ratkaisun kehittäminen perinteisen esiasennusprosessin ti- lalle.
Tutkimuskysymys	Tapaustutkimus: Millaisia työasemahallintatuotteen hallin- taominaisuuksia voidaan tarkastella ja vertailla päätöksen- teon pohjana valittaessa sisäverkkoon sijoitetun ja pilvestä hankitun palvelun välillä? Suunnittelutieteellinen tutkimus: Miten voidaan madaltaa kynnystä tarjota itsepalvelullinen työaseman käyttöönotto pilvipalvelusta tuotettuna?
Tutkijan rooli	Tapaustutkimuksessa tutkijan rooli oli ulkopuolinen tarkkai- lija eli tutkittavaan kokonaisuuteen ei pyritty vaikuttamaan. Suunnittelututkimuksessa tutkijan rooli oli vahvasti osallis- tuva.
Tulokset	Tapaustutkimuksen lopputuloksena saatiin vertailutaulukko Intune-tuotteen hallintaominaisuuksista. Suunnittelututkimuksen lopputuloksena tuotettiin proto- tyyppi sovelluksesta, jolla voidaan suorittaa työaseman käyttöönotto pilvestä itsepalveluna.

3 Teoreettinen viitekehys

3.1 Pilvilaskenta ja pilvipalvelut

Pilvilaskenta (engl. cloud computing) on ylätasoinen käsite internetin yli hajautetusti tarjotuille palveluille. Termillä käsitetään myös uutta palveluiden tuottamisen mallia, johon pohjautuen

tarjotaan sähköisiä palveluita kustannustehokkaasti ja skaalautuvasti *on demand* -periaatteella (Baun ja muut 2011). Liiketoiminnan näkökulmasta pilvilaskennassa on kysymys ulkoistettujen palveluiden tarjonnasta, joita loppuasiakkaat käyttävät samalla tavalla kuin niitä tarjottaisiin yrityksen sisällä omasta ympäristöstä.

Pilvipalvelut-sanalle ei ole yhtä yleisesti hyväksyttyä määritelmää. Pilvipalvelut-termillä tarkoitetaan palvelumallia, jossa resursseja tarjotaan eri tasoilla verkon välityksellä ilman, että käyttäjän tarvitsee ottaa kantaa siihen, missä palvelut fyysisesti sijaitsevat (Salo, 2010). Salo (2010) mukaan kuvattaessa pilvipalveluita yleisesti viitataan yhdysvaltalaisen Nistin (National Institute of Standards and Technology) määrittelemiin pilvipalveluiden ominaispiirteisiin:

- itsepalvelullisuus
- pääsy eri päätelaitteilla
- resurssien yhteiskäyttö
- joustavuus
- käytön mittaaminen (laskutusperuste)

Microsoft puolestaan määrittelee seuraavanlaisia ominaisuuksia pilvipalveluille (Salo, 2010):

- dynaaminen laskentainfrastruktuuri
- palvelukeskeinen lähestymistapa
- käyttöön perustuva laskutus
- automatisoitu hallinta
- itsepalvelu

Pilvipalvelut on palvelumallikäsite, joka ei kuitenkaan määrittele, mistä palveluita tulee tarjota. NIST:n ja Microsoftin määrittelemien ominaisuuksien mukaisia palveluita voidaan tuottaa myös yrityksen sisällä, jolloin yritys on itsessään sekä palveluntarjoaja että asiakas. Tällöin puhutaan yksityisestä pilvestä (engl. private cloud), joka voi sijaita yrityksen omassa konesaliympäristössä tai palveluntarjoajan kapasiteettipalvelussa.

3.1.1 Perusteknologiat

Pilvipalveluiden yksi merkittävä piirre on, että asiakkaan näkökulmasta rakenteen tekninen monimutkaisuus jää piiloon käyttäjän näköpiiristä ja teknologiahaasteet jäävät kokonaan palveluntuottajan ratkaistavaksi. Pilvipalveluiden toinen merkittävä piirre on ratkaisun monistettavuus; karkeasti ilmaistuna kerran luotu voidaan monistaa kaikille. Pilvipalveluiden rakentaminen eri palvelukerroksissa perustuu virtualisointiin, palvelukeskeiseen arkkitehtuuriin ja *www-palveluihin* (Baun, Kunze, Nimis, & Tai, 2011).

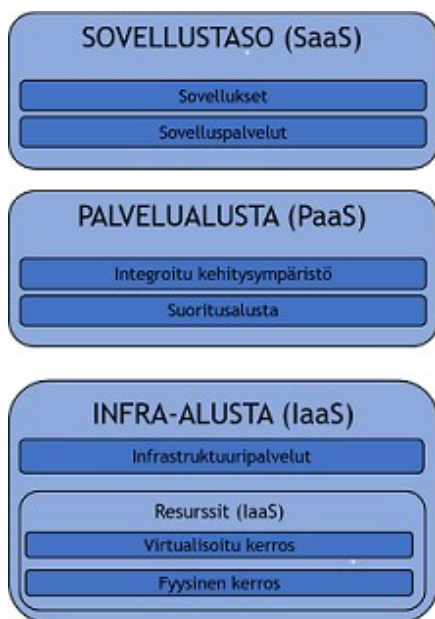
Virtualisointi (engl. virtualization) on pilvipalveluiden perusteknologia, jonka avulla voidaan luoda loogisia kokonaisuuksia siten, että taustalla on vähintään kahdennettuna rakennettu fyysinen laitealusta (Baun et al., 2011). Tyypillisesti asiakkuuksille rakennetaan virtuaalinen verkko, jonka taustalla on fyysisillä lähiverkon kytkimillä toteutettu virtuaalinen lähiverkko (VLAN) ja tähän verkkoon lisätään virtuaalisia palvelinkoneita, joiden taustalla on fyysinen palvelinrypäs (host cluster).

Palvelukeskeinen arkkitehtuuri (engl. service oriented architecture, SOA) on ohjelmistotekniikassa käytetty suunnittelukäytäntö, jonka avulla voidaan integroida riippumattomia ja itsenäisiä, toisistaan erillään suunniteltuja, järjestelmiä toimimaan keskenään saumattomasti eri tasoilla (Sosinsky, 2011). Arkkitehtuurin tavoitteena on pienentää uusien järjestelmien kehittämiseen kuluva aikaa ja toisaalta mahdollistaa vanhojen ratkaisujen hyödyntäminen uusien kehittämisen rinnalla. SOA ei edusta käsitteenä mitään tiettyä ohjelmistoteknologiaa, vaan sillä tarkoitetaan alusta- ja ohjelmointikieliriippumatonta menetelmää jaettujen sovellusten väliseen integraatioon (Sosinsky, 2011).

WWW-sovelluspalvelut (web services) käsittää toisistaan riippumattomia ohjelmistojärjestelmiä, jotka vaihtavat keskenään tietoa käyttäen yhteydenpitoon http- ja https-protokollia. Käsitteellä ei määritellä niinkään käytettäviä teknologioita vaan pikemminkin käyttötarkoitusta ja riippumattomia rajapintoja, joita voidaan yhdistää eri XML-pohjaisia komponentteja käyttäen (Baun et al., 2011).

3.1.2 Pilvipalvelumallit

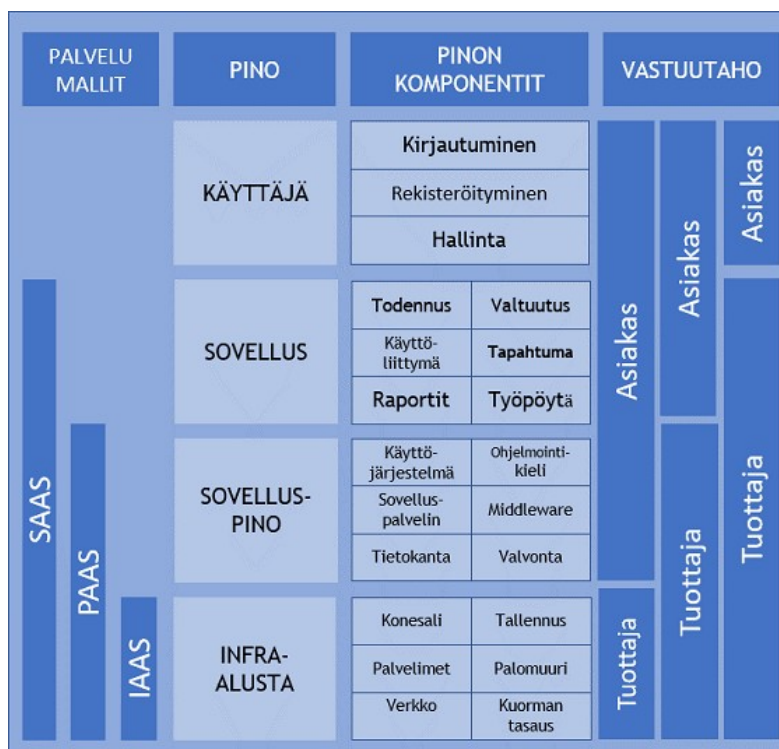
Pilvipalvelumalleja voidaan tunnistaa useita eri yhteyksissä, mutta käytössä on kolme perusmallia (Kuva 3), joilla palvelukokonaisuuksia kuvataan: infrastruktuuri (*IaaS*), palvelualusta (*PaaS*) ja sovellusalusta (*SaaS*) (Baun et al., 2011; Salo, 2010). Oikean palvelumallin tai niiden yhdistelmän valinta on tärkeää ja riippuu aina sovellettavasta kohteesta (Kavis, 2014). Asiakkaan näkökulmasta katsottuna perinteisessä palvelukeskuksessa IT-henkilöstö joutuu rakentamaan alusta lähtien verkko-alustan, palvelinvirtualisointialustan, levypalvelimet ja lisäksi yritys joutuu hankkimaan kaikki tarvittavat lisenssit ennen käyttöönottoa. Kavis (2014) esittää julkaisussaan pinomallisen kaavion palvelutasojen ja osapuolten vastuun vastaavuudesta (Kuva 3).



Kuva 3: Pilvipalvelumallit (Baun et al., 2011).

Palvelumallin alin taso, jossa oman ylläpidon sijaan ostetaan palvelua palvelun tuottajalta, on infrastruktuuri palveluna (engl. *IaaS*, *infrastructure as a service*). Tällä tasolla pilvipalvelussa tarjotaan perustietotekniikkataso, kuten keskitetystä levypalvelimesta varattu tallennustila, verkkoyhteydet ja virtuaalikoneet. Asiakkaalla on tällöin pääsy oman tenanttinsa komponentteihin, esimerkiksi virtuaalikoneisiin, mutta tämän tason alapuolella olevat toiminnot (esimerkiksi uusien virtuaalikoneiden perustaminen) ovat palveluntarjoajan hallinnassa (Kavis, 2014). Tyypillinen tilanne on sellainen, jossa asiakkaalle perustetaan IaaS-palvelumallissa yksityinen pilvi (engl. *private cloud*), joka tarkoittaa esimerkiksi yhtä IP-aliverkkoa sisältäen kaikki kyseisen asiakkuuden virtuaalipalvelimet.

IaaS-tasolla voidaan myös kuvata yrityksen sisäistä konesalirakennetta, jolloin kyse ei ole kaupallisesta toiminnasta moniasiakkuusympäristössä vaan puhtaasti omaan käyttöön rakennettu ympäristö eli kuvan 3 kaaviossa toimittajan (engl. *vendor*) ja asiakkaan (engl. *customer*) vastualueet kuuluvat samalle toimijalle. Tämä rakenne tuottaa yritykselle samat hyödyt kuin palveluntarjoajallakin, mutta perustamiskustannukset ovat yhtä yritystä ajatellen huomattavan suuret, vaikkakin käytännön operointi eli resurssien käyttöönottoaminen tai niiden vapauttaminen on erittäin dynaamista, kustannustehokasta ja nopeaa. Kuvattua yrityksen sisäistä IaaS-alustaa kutsutaan yksityiseksi pilveksi (engl. *private cloud*), jonka käyttöönottoaminen mahdollistaa jatkossa palveluiden kokonaan tai osittaisen siirron palveluntarjoajan ympäristöön virtuaalisten resurssien mahdollistamaa joustavuutta hyödyntäen. Virtuaaliresurssien jakamista omaan yksityiseen sekä palveluntarjoajan (*public cloud*) pilviympäristöön kutsutaan käyttöönottomallina hybridiratkaisuksi (engl. *hybrid cloud*) (Salo, 2010).



Kuva 4: Cloud Stack (Kavis, 2014).

Palvelupinossa seuraavana tasona on sovellusalusta palveluna (engl. platform as a service, PaaS), jota asiakkuudet voivat hyödyntää esimerkiksi sovellusten kehittämisessä, ylläpidossa ja testaamisessa (Salo, 2010). Palvelun hyödyllisyys realisoituu siten, että sovelluskehittäjän ei tarvitse ottaa kantaa itse infrastruktuurin rakentamiseen vaan se on saatavilla valmiina hyödynnettävinä moduuleina käyttöönotettavaksi. Hyötyjen, kuten, nopeus, kustannustehokkuus ja skaalautuvuus, lisäksi PaaS-mallissa on Salo (Salo, 2010) mukaan riskinä lukittuminen tiettyyn palveluntarjoajaan, mikä saattaa edellyttää esimerkiksi tiettyä ohjelmointikieltä alustaa hyödynnettäessä. Tällöin pullonkaulaksi ja käytännön riskiksi muodostuu tekijöiden konkreettinen osaaminen huomioiden työmarkkinat ja osaamisprofiilit, joita on kunakin ajankohtana saatavilla.

PaaS-alustan ja palveluntarjoajan valinta luo strategiselle johdolle haasteita pitkän tähtäimen suunnittelussa valittaessa ekosysteemiä, jonka päällä toimitaan. Pinomallin IaaS-tasolla käytettävät standardit ja käytännöt ovat vakiintuneet ajan saatossa siten, että siirtyminen palveluntarjoajien välillä on realistisesti mahdollista ja joustavaa kohtuullisin kustannuksin. IaaS-tason (Kuva 4) osista erityisesti tietoliikenne (engl. firewall ja network)- sekä tallennus (storage)-komponentit ovat valmistajariippumattomasti hyvin standardoituja. Käytännössä tällä tasolla ympäristöstä toiseen siirtyminen tarkoittaa virtuaalikoneiden siirtämistä ja mahdollisesti muuntamista sopivaksi kohdealustalle sopivaksi. Reaalimaailmassa käytettävät virtuaali-

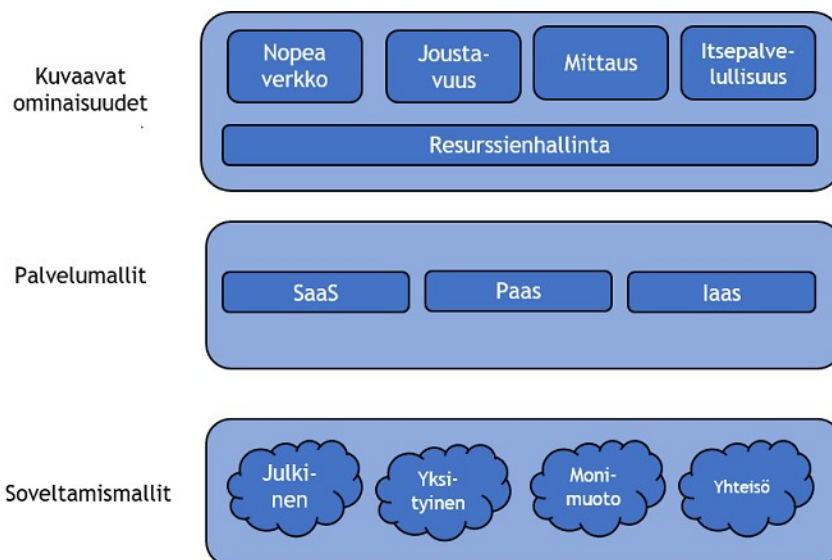
konealustat ovat valtaosin joko Microsoftin Hyper-V- tai VmWaren ESX-tuotteisiin pohjautuvia, joiden välillä pystytään toteuttamaan muunnoksia täydellisesti.

Pinon ylimpänä palvelukomponenttina on sovellukset palveluna (engl. software as a service, SaaS), jolloin asiakas ostaa omistamisen, asentamisen ja ylläpidon sijaan tarvittaessa-periaatteella (Salo, 2010). Tällöin siis kuvan 3 mallissa asiakas ei enää ota kantaa edes sovellusten ylläpitoon tai siihen millaisella alustalla sovelluksia tarjotaan, vaan sen sijaan haluttuja sovelluksia hankitaan esimerkiksi käyttäjämääräperusteisesti käytettäväksi internetin yli. Näin toimimalla sidottu pääoma saadaan mahdollisimman pieneksi ja puhutaan käytännössä ainoastaan muuttuvista kustannuksista. Tyypillisesti yritykset ostavat SaaS-palveluna sovelluksia, jotka eivät liity suoranaisesti ydinliiketoimintaan vaan pikemminkin tukitoimintoihin kuten taloushallinto, henkilöstöhallinto ja toiminnanohjaus (Kavis, 2014).

Palveluntarjoajan näkökulmasta katsoen sovelluksen tarjoaminen SaaS-palveluna moniasiakkuusperiaatteella (multitenancy) antaa mahdollisuuden mahdollisimman kustannustehokkaiseen (Salo, 2010) ja vähän ylläpitoa vaativaan toimintamalliin. Esimerkkinä tällaisesta palvelusta voidaan mainita esimerkiksi pilvipalveluna tarjottava Gmail-sähköposti, jossa samaa palvelua tarjotaan yhdestä ympäristöstä ainoastaan lisäämällä uusia käyttäjiä sen sijaan, että jokaiselle asiakkuudelle perustettaisiin oma sähköpostipalvelininstanssi IaaS-tasolla. Samanlaisesti kuitenkin haasteeksi tulee kuitenkin monen asiakkuuden ympäristössä kultaisten keskien löytäminen; miten paljon luodaan yhteistä pohjaa ja toisaalta miten paljon asiakkuuskohdaisesti on mahdollista tehdä räätälöintejä (Salo, 2010)? Mallin IaaS-tasolla ongelmia ei ole, koska kapasiteetin lisääminen taustalle sovellusvaatimusten ja asiakasmäärien kasvaessa voidaan vastaavasti laitealustaa laajentaa dynaamisesti ja läpinäkyvästi.

3.1.3 Käyttömallit

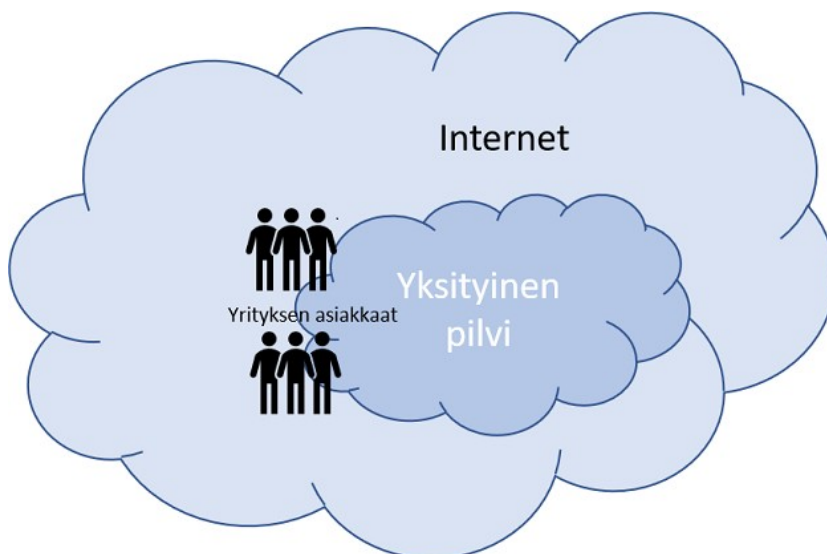
Käyttömallilla kuvataan perusrakenne, jolla pilvipalveluita tarjotaan asiakkaille. Käytettävät tekniset ratkaisut puolestaan määräytyvät sen mukaisesti, mihin palveluiden eri osat sijoitetaan sekä miten eri tasojen välinen liikennöinti ja pääsynvalvonta toteutetaan (Waschke, 2012). Käyttömallit voidaan jakaa neljään perusmalliin, joita ovat julkinen (public), yksityinen (engl. private), hybridi (engl. hybrid) ja yhteisökäyttöinen (engl. community) pilvi (Kuva 5) sen mukaan, millä toimijalla on hallintaoikeus palvelukokonaisuuteen (Kavis, 2014).



Kuva 5: Pilvilaskennan yleiskuva (Kavis, 2014).

Yksityinen pilvi (engl. private cloud) on palvelusisältö, joka on yhden yrityksen tai muun toimijan hallinnoima kokonaisuus. Palvelukokonaisuus voi sijaita asiakkaan omassa konesaliympäristössä, mutta tyypillisesti sijoituspaikkana on palveluntarjoajan palvelinkeskus (Waschke, 2012), jossa asiakkuudelle on lohkaistu IaaS-tasolla oma sisäinen IP-aliverkko, mihin asiakkuuden virtuaalipalvelinkoneet on sijoitettu. Asiakas pääsee itse hallinnoimaan kyseistä kokonaisuutta itsenäisesti aivan, kuten kyseinen ympäristö olisi sijoitettuna fyysisesti yrityksen hallinnoimaan palvelinkeskukseen.

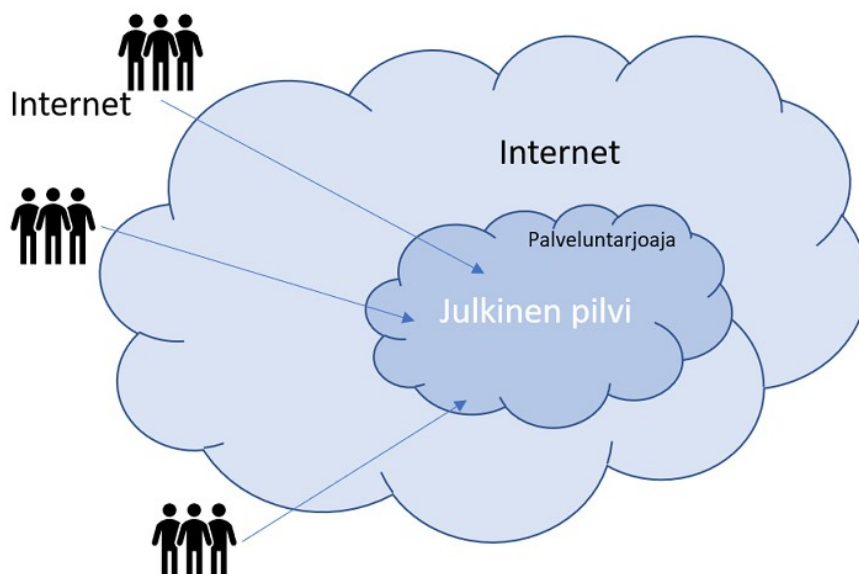
Yksityisen pilven käyttömallissa toimijat jaetaan Waschken (2012) mukaan kuluttajiin ja palveluntuottajiin (Kuva 5). Tapauksissa, joissa pilvi sijoitetaan asiakkaan omaan fyysiseen konesaliin, toimii palveluntuottajana yksiselitteisesti yrityksen oma tietohallinto-osasto ja kuluttajina yrityksen muiden toimintojen loppukäyttäjät. Silloin, kun yrityksen yksityinen pilvi tuotetaan IaaS-tasolla palveluntarjoajan kapasiteettipalvelussa toimijamalli (Kuva 6) säilyy samana, mutta fyysisen alustan hallinnointivastuu siirtyy palveluntarjoajalle, jolloin asiakkaan tietohallinto on tältä osin kuluttajan roolissa. Fyysiseen alustaan sisältyy tässä tapauksessa tärkeänä elementtinä palomuri (Waschke, 2012), jonka hallinnointi on yksityisen pilven reunalla palveluntarjoajan vastuulla. Tyypillisesti tällaisessa ratkaisussa, jossa yksityinen pilvi tarjotaan palveluntarjoajan tuottamana, luodaan asiakkaan fyysisten toimipisteiden ja pilven väliin suojattu site-to-site VPN-tunneli.



Kuva 6: Yksityisen pilven toimijamalli (Waschke, 2012).

Julkinen pilvi (engl. public cloud) on arkkitehtuuriltaan yksityistä pilveä vastaava, mutta ne eroavat toisistaan siten, että julkisen pilven mallissa palveluita tarjotaan julkisesti kaikkien palveluntarjoajan asiakkaiden kesken (Waschke, 2012). Käytännössä julkista pilveä tarjotaan sovellustasolla (SaaS), kun taas yksityistä pilveä tuotetaan yleensä laaS-tasolla eli tarjotaan palvelinalusta yksittäiselle asiakkaalle. Julkisessa pilvessä SaaS-tasolla tuotetut palvelut voivat olla esimerkiksi palkkahallinto-, CRM- tai ERP-sovelluksia, joita tarjotaan moniasiakkuusperiaatteella eli multitenantisti.

Waschken (Waschke, 2012) mukaan julkinen pilvi tulisi korvaamaan tulevaisuudessa yksityisen pilven asteittain, mutta tämä ei liene mahdollista ainakaan nykyisen ajattelumallin mukaisesti, missä jokainen yritys haluaa pitää oman ympäristönsä eristettynä muista. Lisäksi, kun julkisesta pilvestä ostetaan palveluita esimerkiksi SaaS-tasolla, päädytään helposti sirpalemaiseen ratkaisuun, jossa yksittäiset käyttäjät kukin erikseen ottavat yhteyttä pilveen omista toimitiloistaan (kuva 6). Tällaisessa mallissa esimerkiksi käyttäjätodennuksen (authentication) yhtenäistämässä voi tulla haasteita varsinkin, kun nykyisin tyypillinen todennusratkaisu on yrityksen sisäinen Microsoftin aktiivihakemisto, jonka ohjauspalvelimet (domain controller) on sijoitettu fyysisesti yrityksen tiloihin tai palveluntarjoajan tuottamaan yksityiseen pilveen. Käytettäessä pelkkiä julkisen pilven palveluita tarvitaan tällöin myös SaaS-tasolla tarjottava moniasiakkuuteen eli multitenanttsuuteen soveltuva hakemistoratkaisu (aktiivihakemisto tai muu LDAP:iin perustuva), mitä vastaan muut SaaS-sovellukset voivat todentaa käyttäjän.

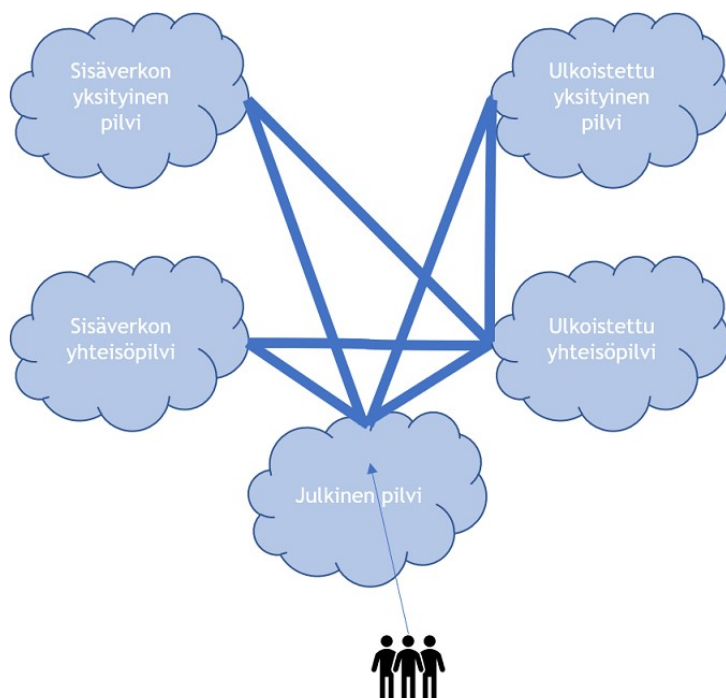


Kuva 7: Julkisen pilven toimijamalli (Waschke, 2012).

Yhteisöpilvi (engl. community cloud) on eri toimijoiden hyödyntämä kokonaisuus määrätyistä toiminnoista (Waschke, 2012), joista on muodostettu yhteinen rajattu palvelu. Tämän lisäksi kaikilla toimijoilla voi olla rinnalla esimerkiksi yksityinen pilvi omia sisäisiä peruspalveluita varten ja samanaikaisesti tiettyjä sovelluksia voidaan ostaa julkisesta pilvestä SaaS-palveluna. Muiden käyttömallien tavoin yhteisöpilvi ei itsessään edusta mitään erillistä teknologiaa vaan käsite pikemminkin kuvaa rajatun palvelun käyttötarkoitusta eri tahoilta tulevien toimijoiden näkökulmasta (Waschke, 2012). Tyypillinen esimerkki Suomessa käytössä olevasta yhteisöpilvi-tyyppisestä palvelusta on Sosiaali- ja terveystieteiden ministeriön koordinoima Kanta-palvelut (Kansallinen Terveysarkisto (Kanta), 2015), jonka toimijaosapuolina ovat julkiset ja yksityiset lääkärit, apteekit ja yksityishenkilöt.

Hybridipilvi (engl. hybrid cloud) kuvaa erilaisia yhdistelmiä muista käyttömalleista (yksityinen, julkinen, hybridi) (Waschke, 2012), joista muodostetaan tarpeen mukaisia kokonaisuuksia. Tyypillinen esimerkki hybridimallista on yrityksen sisäinen verkko, josta osa sijoitetaan omiin tiloihin ja osa palveluntarjoajan IaaS-mallin mukaiselle alustalle. IP-tasolla kaikki palvelut ovat loogisesti samassa IP-verkkoavaruudessa. Hybridipilvi käsitteenä on niin laaja ja yhdistelmämahdollisuuksia on lukemattoman paljon, joten ei ole realistista mallintaa kaikkia eri kokonaisuuksia (Badger, Grance, Patt-Corner, & Voas, 2012). Badger et al. (Badger et al., 2012) kuvaavat mahdollisia hybridipilviyhdistelmiä Kuva 8, joissa osapuolina ovat eri sijainneissa olevat yksityiset pilvet ja yhteisöratkaisut sekä julkinen pilvi. Näistä varsinkin yhteisöpilven ja hybridipilven ero itsessään voi olla haasteellista, ellei mahdotonta erotella tietyissä ratkaisukokonaisuuksissa. Lisäksi mallin tunnistamiseen vaikuttaa kunkin tarkastelijan valitsema näkökulma.

Hybridimallin mukaista toimintaympäristöä voidaan hyödyntää siten, että pysyvät perustoinnallisuudet sijaitsevat yksityisessä pilvessä, mutta väliaikaisesti voidaan ostaa kapasiteettia julkisesta pilvipalvelusta. Tällaisia käytännön tarpeita ovat esimerkiksi väliaikainen lisätalennustila ja dynaamiseen kuormantasaukseen tarvittavat edustapalvelimet oman www-pohjaisen palvelun julkaisemisessa yleiseen internetiin.



Kuva 8: Hybridimalli (Badger et al., 2012).

3.2 Multitenanttisuus

3.2.1 Määritelmiä

Englanninkielinen sana tenant suomennetaan vuokralaiseksi eli asiakkaaksi, joka vastikkeellisesti saa käyttöönsä sovituksi ajaksi halutun resurssin. Etuliite multi tarkoittaa useaa vuokralaista tai asiakasta ja tästä johdettuna sana multitenantti (engl. multi tenant) voidaan suomentaa tutkimuksen aihepiirin yhteydessä moniasiakkuudeksi tai multitenanttisuudeksi. Termi vuokralainen tuo yleensä mieleen esimerkiksi asunnon tai vastaavan tilan vuokraamisen ja tästä johtuen on tarkoituksenmukaista käyttää tutkimuksen aihepiirin yhteydessä käsitteitä asiakkuus ja moniasiakkuus sekä moniasiakkuusarkkitehtuuri tai multitenanttisuus.

ISO:n (ISO, 2014) mukaan moniasiakkuusarkkitehtuurille (engl. multi-tenancy) on ominaista fyysisten ja virtuaalisten resurssien jakaminen usean asiakkuuden (tenant) kesken siten, että näiden käyttämä laskenta-alusta ja tallennettu tieto eristetään omiksi suljetuiksi kokonaisuuksiin. Tyypillisesti moniasiakkuusympäristössä samaan organisaatioon kuuluvat pilvipalve-

lun käyttäjät muodostavat yhden kokonaisuuden eli asiakkuuden. Tilanteessa, missä käyttäjien ryhmä, tenantti, muodostuu useaan eri organisaatioon kuuluvista käyttäjistä, liittyy moniasiakkuusympäristön yhteisöpilvi- tai julkinen pilvi -käyttömalleihin. Lisäksi sama asiakasorganisaatio voi muodostaa useita asiakkuuksia eli tenantteja saman pilvipalvelutarjoajan kaupallisena asiakkuutena, mutta organisaation rakenteen vuoksi on perusteltua erotella esimerkiksi eri tytäryhtiöt tai osastot omiksi eristetyiksi ympäristöikseen.

Tutkimusyhtiö Gartner (Gartner, IT Glossary) määrittelee englanninkielisen termin multi-tenancy ja siihen liittyvät muut käsitteet kahden eri skenaarion näkökulmasta. Ensimmäisessä skenaariossa tenantit (asiakkuudet) ovat sovellusinstansseja, joita suoritetaan loogisesti toisistaan eristettynä jaetussa fyysisessä ympäristössä. Toisessa skenaariossa tenantit ovat organisaatioita, joilla on käyttöoikeus jaettuun sovellukseen. Gartner esittää myös kolmantena skenaarionaan ensimmäisen kanssa päällekkäin menevän näkemyksen, jonka mukaan sovellukset kilpailevat jaetusta resurssista. Määritelmä ei ota kantaa pilvipalvelumallin tasoihin ja määritelmä viittaa lähinnä ylimpään SaaS-sovellustasoon, kun taas ISO:n mukaan myös fyysisten resurssien jakamista jaetulla alustalla kutsutaan multitenanttisuudeksi. Lisäksi Gartner on määritellyt erikseen käsitteen joustava multitenanttisuus (engl. elastic multi-tenancy) (Gartner, IT Glossary), jonka sisältö on käytännössä samanlainen kuin käsitteen multi-tenancy.

NIST (Badger et al., 2012) määrittelee multitenanttisuuden tehokkaaksi palveluntarjoajan resurssien jakamiseksi, ja tuo myös esille menetelmiä multitenanttisuuden toteuttamiseksi pilvipalvelumallin eri tasoilla. IaaS-tasolla kysymys on virtuaalikoneiden perustamisesta jaetun fyysisen laitealustan päällä, PaaS-tasolla prosessit jakavat saman käyttöjärjestelmän ja verkkoalustan tarjoamat resurssit ja SaaS-tasolla pilvipalvelun käyttäjät jakavat saman sovellusinstanssin sekä siihen mahdollisesti liittyvän tietokannan eri asteisesti. Badger et al. (Badger et al., 2012) tuovat esille myös multitenanttisuuteen liittyvät tietoturvariskit esimerkiksi asiakkuuksien eristämisen yhteydessä tapahtuvissa virheissä, jolloin asiakkuuksille voi tarjoutua pääsymahdollisuus toisen yrityksen tietoihin. Tietovuotoihin liittyvät tutkimusongelmaan tarjotaan ratkaisuvaihtoehtona tiedon kryptaamista ennen pilvipalveluun tehtävää tiedontalennusta.

3.2.2 Miten multitenanttisuutta voidaan lähestyä?

NIST:n määritelmän (Grance & Mell, 2011) mukaan pilvipalveluissa esiintyy viisi perustavaa laatua olevaa ominaisuutta, joista yhtenä multitenanttisuuteen liittyvänä piirteenä on resurssien jakaminen (resource pooling) (sivu 8.), millä tarkoitetaan palveluntarjoajan fyysisten ja virtuaalisten resurssien jakamista kaikkien asiakkuuksien kesken käyttäen moniasiakkuusarkkitehtuurimallia (engl. multi-tenant model) hyödyntämällä käytössään olevaa käyttöalustaa ja

muuttamalla asiakkuuden käytössä olevien resurssien määrää tarpeen mukaisesti ja kustannustehokkaasti. Asiakas voi tällaista palvelua hyödyntäessään vaikuttaa käytettävän palvelun sijaintiin ainoastaan isommassa mittakaavassa, kuten esimerkiksi valtio-, maanosa tai palvelinkeskustasolla (Grance & Mell, 2011).

Resurssit voidaan ottaa joustavasti (rapid elasticity) (Grance & Mell, 2011) käyttöön tai vapauttaa käyttötarpeen ja tilanteen muuttuessa. Resurssien käyttäjän näkökulmasta kapasiteettia on saatavilla rajattomasti ajankohdasta riippumatta.

Moniasiakkuusarkkitehtuuria voidaan lähestyä Bojanovan (Bojanova, Fiaidhi, Zhang, & Zhang, 2012) mukaan kolmella eri ratkaisumetodilla riippuen, millä tasolla asiakkuuksia halutaan erottaa toisistaan: tietokannan, virtualisoinnin tai fyysisen eristäminen muodossa, joista voidaan vaatimusten mukaan käyttää tarvittavia menetelmiä ja niiden yhdistelmiä eri pilvipalvelumallin tasoilla (taulukko 1).

Taulukko 2: Multitenanttisuuden toteutus pilvipalvelutasoilla.

PILVIPALVELUTASO	JAETTU RESURSSI	METODI	KÄYTTÄJÄ	KUSTANNUSTASO
SaaS	Sovellus, tietokanta	Tietokanta, sovellusasetukset	Käyttäjät eri organisaatioista	Edullisin
PaaS	Käyttöjärjestelmä, kehitysalusta		Sovellukset, prosessit	
IaaS	Fyysinen alusta	Virtualisointi	Virtuaalikoneet	
Fyysinen	Ei jaettua resurssia	Fyysinen erottaminen		Kallein

3.2.3 Jakaminen vs. eristäminen

Multitenanttisuuden yhteydessä liikutaan janalla, jonka vastakkaisissa päissä ovat resurssien jakaminen ja tietoturvallisuuden piiriin liittyvä asiakkuuksien eristäminen omiin virtuaaliin ympäristöihinsä eli yksityisiin pilviin. Toisessa päässä on pilvipalveluiden ekonomiassa oleellisenä lähtökohtana tuottaa kustannustehokkaasti palveluita toisistaan riippumattomille asiakkuuksille ja toisessa päässä vaatimus turvata kaikkien tenanttien tiedon yksityisyys samojen resurssien jakamisesta huolimatta (Rymer & Staten, 2012).

Pilvipalveluita tarjoavien yhtiöiden tulee resurssien tehokkaaseen jakamiseen perustuvassa liiketoiminnassa huomioida suunnitteluvaatimukset toisaalta uusille sovelluksille ja taas toisaalta olemassa oleville sovelluksille multitenanttisuuden näkökulmasta (Rymer & Staten, 2012). Sovelluksen sijoittaminen pilvipalveluun SaaS-tasolla multitenantisti toisistaan riippumattomille asiakkuuksille edellyttää näiden asioiden huomioimista esimerkiksi tietokanta- ja konfiguraatiotasolla. Uutta sovellusta kehitettäessä multitenanttisuus voidaan huomioida jo suunnittelulähtökohtana, mutta jo käytössä olevan tuotteen istuttaminen multitenanttiin

käyttöön pilvipalvelussa usein vaatii muutostyötä sovelluksen tietokannan ja konfiguraatietiedostojen rakenteeseen. Tilanne, jossa sovelluksen muutostyötä ei voida toteuttaa, ei myöskään multitenanttisuus toteudu sovellustasolla vaan tällöin ratkaisuna on perustaa asiakkuuksittain palvelininstansseja infrastuktuuritasolla, jolloin multitenanttisuus siirtyy näin SaaS-tasolta laas-tasolle ja samanaikaisesti palvelun tuottaminen on ratkaisuna lähtökohtaisesti kalliimpaa sekä asiakkuuden perustamisen että elinkaaren hallinnan vaiheissa.

3.2.4 Multitenanttisuus infrastruktuuritasolla

Infrastruktuuri- eli laaS-palvelutasolla multitenanttisuus toteutetaan käyttämällä metodina virtualisointia virtuaalikoneiden, virtuaalisten levyasemien ja virtuaaliverkkojen muodossa. Alusta, mistä virtuaaliset resurssit varataan, on fyysinen ympäristö, joka sijaitsee asianmukaisessa konesaliympäristössä joko yrityksen omassa tai palveluntarjoajan palvelukeskuksessa. laaS-palvelun elementit, joista palvelu muodostuu, ovat palvelimet, levypalvelimet ja kytkimet ja näistä rakennetaan vikasietoinen vähintään kahdennettu fyysinen ympäristö. Asiakkaan näkökulmasta näkyvin elementti on palvelinalusta, mihin perustetaan haluttu määrä virtuaalikoneita käyttöjärjestelmineen, joiden päällä liiketoimintaympäristön sovelluspalvelut tuotetaan. Toisena elementtinä laaS-ympäristössä tarvitaan tallennustilaa, joka tuotetaan useimmiten verkkoon liitettyllä erillisellä tallennuspalvelimella (engl. NAS, network attached storage), mistä virtuaalikoneisiin liitetään nopean, dedikoidun tallennusverkon välityksellä virtuaalisia kiintolevyjä. Kolmantena elementtinä on fyysisellä kytkinverkolla tuotetut virtuaaliset verkkoyhteydet, jotka eristetään asiakkuuksittain (engl. tenant) toisistaan virtuaalisilla lähiverkoilla (VLAN).

Multitenanttisuus saavutetaan tällä palvelutasolla siten, että asiakkuudet eristetään verkkotasolla VLAN-verkoilla, jolloin jokainen asiakkuus eli tenantti eristetään omaan yksityiseen pilveensä (engl. private cloud), mihin luodaan uusia virtuaalikoneita tai tuodaan valmiit virtuaalikoneet toisesta tuotantoympäristöstä. Virtuaalikoneiden käyttöönottovaiheessa niihin yhdistetään verkkotallennuspalvelimelta tarvittava määrä virtuaalisia levyasemia, joihin kokoa voidaan kasvattaa tarpeen mukaan. Edellä kuvatulla tavalla palveluntarjoajan näkökulmasta otetaan kaikille asiakkuuksille eli tenanteille samalla ylläpitoprosessilla nopeasti ja joustavasti käyttöön juuri tarvittavat määrä kapasiteettia, jolloin fyysisen alusta käyttö saadaan mahdollisimman optimaaliseksi ja kustannustehokkaaksi. Asiakkaan näkökulmasta palvelu näkyy nopeana ja joustavana sekä skaalautuvana palveluna.

3.3 Multitenanttisuus sovellustasolla

Multitenanttia sovellusta käyttävät asiakkuudet jakavat sovelluksen resursseja siten, että asiakkuudelle sovellus näkyy niin kuin kyseessä olisi vain omaan käyttöön suunniteltu käyttöympäristö. Asiakkuudet eivät saa nähdä toisiaan millään tasolla. Asiakkaan eli tenantin näkökulmasta tuotteelle asetettavat vaatimukset voivat vaihdella, mutta seuraavia ominaisuuksia voidaan yleisesti pitää multitenantin sovelluksen perusominaisuuksina (Betts, Homer, Jezierski, Narumoto, & Zhang, 2012):

1. Eristäminen (engl. isolation) eli tenanttien toiminnot eivät saa näkyä tai vaikuttaa toisiinsa samoin kuin asiakkuuksien tietovarannot tulee pitää luotettavasti toisistaan erillään.
2. Saatavuus (engl. availability) on asiakkuuksien näkökulmasta palvelutasosopimusten mukaan määriteltyä saatavuustasoa, joka on käytännössä tänä päivänä yli 99 %. Saatavuuden varmistaminen on palveluntarjoajan tärkein tehtäviä huomioimalla mm. laitealustan ja muun konesaliympäristön vikasietoisuus kaikilla tasoilla sekä resurssien kuormituksen seuraaminen ja ylikuormitustilanteisiin reagoiminen ennakoivasti hankkimalla ajoissa lisää kapasiteettia.
3. Skaalautuvuus (engl. scalability) on käyttäjätahon näkökulmasta tärkeää ja sovelluksen suunnittelussa onkin tärkeää määritellä, mitkä ominaisuudet ovat kaikille asiakkuuksille yhteisiä ja toisaalta, mitkä ominaisuudet ja missä rajoissa asiakkuuksien itse tekemät määritykset ovat mahdollisia. Tässäkin on huomioitava mahdolliset vaikutukset muihin eristettyihin tenantteihin.
4. Kustannukset (engl. costs) on oleellinen liiketoiminnallinen elementti multitenantin sovelluksen myymisessä ja ostamisessa. Resurssien jakaminen sovellustasolla alentaa asiakkuuksien kustannuksia, koska perustamiskuluja ei juurikaan ole ja tämän lisäksi omien palvelininstanssien omistaminen ja ylläpitäminen jää kokonaan pois. Palvelu-toimittajan näkökulmasta sovellus luodaan ja alusta perustetaan vain kerran ja ylläpito kohdistuu vain yhteen palvelininstanssiin, jolloin kaikille asiakkuuksille tehdään ainakin tiettyjen elementtien osalta asiat vakioidusti ja kontrolloidusta.
5. Räätelöitävyys (engl. customizability) on piirre, jota voidaan toteuttaa eri tasoilla riippuen sovelluksen ominaisuuksista ja asiakkaiden tarpeista. Nämä asiat tulee huomioida sovellusympäristön suunnitteluvaiheessa hyvin tarkasti, koska jälkikäteen usean toimijan ympäristössä tehtävät perustavanlaatuiset muutokset realisoituvat usein erittäin haasteellisina toimenpiteinä.

6. Soveltuvuus erilaisten säännösten (engl. regulatory compliance) kanssa voi olla haasteellista riippuen asiakkuuden toimialasta ja pilvipalvelun sijaintivalttiosta. Varsinkin viranomaistahot noudattavat esimerkiksi tietosuojalain vaatimusta henkilötietojen sijainnista, mikä sulkee pois ulkomaisia vaihtoehtoja palveluntarjoajan valinnassa.

3.3.1 Sovellusarkkitehtuuri

Sovellus, jota tarjotaan palveluna eli SaaS-palveluna, määritellään internetin yli palveluntarjoajan tuottamaksi sovellukseksi. Tämä ei itsessään kerro mitään siitä, onko sovellus suunniteltu moniasiakkuusympäristö huomioiden vai onko kyseessä esimerkiksi vanha tuote, jossa jokaiselle asiakkaalle tarjotaan oma erillinen sovellusinstanssi. Tämä on oleellista kustannusten ja edelleen tuotteelle määräytyvän hintalapun kannalta.

Chongin & Carraroron (Chong & Carraro, 2006) mukaan sovellusarkkitehdin näkökulmasta voidaan määritellä kolme tärkeää tekijää, joiden avulla arvioidaan, miten hyvin rakennettu sovellusympäristö soveltuu SaaS-käyttöön:

(1) *Skaalautuvuudella* tarkoitetaan tehokasta resurssien yhtäaikaista käyttöä asiakkuuksien kesken samassa sovellusympäristössä.

(2) *Multitenanttisuuteen* soveltuvuus on ehkä painavin tekijä (Chong & Carraro, 2006) eroteltaessa toisistaan yhdelle asiakkuudelle soveltuvaa tai monen asiakkaan käyttöön soveltuvaa sovellusta. Moniasiakkuuteen eli multitenanttisuuteen soveltuva tuote tarjoaa saman ohjelmistoperustan kaikille asiakkuuksille niin, että asiakkuuksien tiedot on turvallisesti eroteltu toisistaan. Tämä kuitenkin on pystyttävä tuottamaan siten, että palvelu toimitetaan yhdestä instanssista.

Saman ohjelmistoperustan lisäksi multitenantissa sovelluksessa asiakkuuksille voidaan tehdä erilaisia räätälöintejä puuttumatta itse ohjelmiston ydinkoodiin. Multitenantissa sovelluksessa asiakaskohtainen (3) *konfiguroitavuus* pystytään tuottamaan käyttämällä metatietoja (Chong & Carraro, 2006) eli vakioituja asetuskenttiä, joiden avulla asiakkuudet voivat tehdä ennalta määriteltyjä asioita hallitusti. Aidossa jaetussa sovelluksessa onkin tyypillistä se, että asiakkuudet toimivat tiettyjen rajojen sisällä, jolloin mitä hyvänsä räätälöintejä ei voida toteuttaa.

3.3.2 Pilvisovelluksen valmiusasteet multitenanttiin käyttöön

Sovelluksen valmiusaste, kypsyys tai toisin ilmaistuna maturiteetti multitenanttiin käyttöön voidaan Chongin & Carraron (2006) mukaan ilmaista näiden kolmen attribuutin mukaan, joiden orjallinen noudattaminen ei ole itsetarkoitus, sillä tietty sovellus voi vastata vain osaa näiden määreiden vaatimuksista ja silti soveltua liiketoiminnan käyttöön hyväksyttävällä tasolla. Valmiusastetta voidaan pitää pikemminkin kuvan (Kuva 9) tapaisena jatkumona, jossa sovellus asettuu ääripäiden välille tasapainoiseen asemaan.



Kuva 9: Eristäminen vs. jaettu ympäristö.

Taso 1

Sovelluksesta perustetaan jokaiselle asiakkuudelle oma palvelininstanssi, johon sijoitetaan kyseisen asiakkuuden sovellus ja siihen liittyvä tietovarasto. Tällöin kyseessä on perinteinen ASP-sovellusvuokrausmalli, joka oli yleisesti käytössä 1990-luvulla (Chong & Carraro, 2006), eikä tässä tapauksessa voida puhua sovellustason multitenanttisuudesta vaan kysymys on pikemmin yksityisten pilvien perustamisesta infrastruktuuritasolla.

Tällä tasolla voidaan suoraan virtualisoida esimerkiksi yrityksen turvattomaksi todetussa toimitilassa sijaitseva fyysinen palvelin sellaisenaan palveluntarjoajan konesaliin, johon on perustettu asiakkuutta varten oma yksityinen pilvi.

Taso 2

Toisella tasolla jokaisella asiakkuudella on edelleen oma instanssi samalle sovellukselle, mutta pohjana käytetään palveluntarjoajan luomaa peruspohjaa (Chong & Carraro, 2006) instanssin perustamiselle ja tarvittaessa asiakkuuskohtaisesti voidaan muokata räätälöityjä muutoksia tarpeen mukaisesti.

Ensimmäinen valmiustaso on tyypillinen vanhojen fyysisten yksittäisten sovellusten siirtämisessä asiakkaan tiloista palveluntarjoajan palveluun, kun taas toinen taso on mallinnettu alusta, jossa asiakkuuksille rakennetaan alusta lähtien sovellusalusta ja tällöin kyseistä sovellusta myydään tuotteistettuna sovelluspalveluna (SaaS) asiakkuuksille.

Taso 3

Kolmannella tasolla on käytössä yksi palvelininstanssi, josta tarjotaan kaikille multitenantisti asiakkuuksille samaa palvelua siten, että asiakkuudet eivät saa näkyvyyttä toisiinsa millään tasolla.

Taso 4

Neljäs taso on loogisesti samanlainen kolmannen tason kanssa, mutta mukaan tuodaan skaalautuvuus eli palvelininstanssi on loogisesti yksi osoite, mutta taustalla sovellusympäristö toimii klusteroidusti ja vikasietoisesti käyttäen alustana kahta tai useampaa palvelinta, jotka voivat olla virtuaalisia tai fyysisiä koneita.

3.3.3 Metatieto

Pilvipalveluun soveltuvassa ohjelmistossa metatiedolla pystytään tarjoamaan asiakkuuksille rajapinta erilaisten omien konfiguraatioiden ja räätälöintien tekemiseen. Tällä tavalla ydinohjelmisto on kaikille yhteinen ja räätälöintirajapinta eli metatiedon käyttäminen on omalta osaltaan mallinnettu ja testattu tapa tehdä vakiomuunnoksia sovelluksen eri osa-alueilla. Tyypillisiä osa-alueita metatiedon hyödyntämisessä ovat käyttöliittymä ja yrityskohtainen brändäys, yrityksen työprosessien ohjaus, muut laajennukset tietokantatauluihin ja pääsynhallinnan ohjaus (Chong & Carraro, 2006). Metatiedon käyttömahdollisuutta voidaankin pitää yhtenä oleellisena tekijänä tarkasteltaessa sovelluksen kypsyyssastetta multitenanttiin käyttöympäristöön.

3.3.4 Todennus ja valtuutus

Palveluntarjoaja voi järjestää sovelluksen käyttäjähallinnan joko keskitetysti tai delegoida hallintavastuun tenanteittain eli asiakkuuskohtaisesti suoraan asiakkaalle. Keskitetty hallinta aiheuttaa enemmän työtä päivittäisten palvelupyyntöjen muodossa, kun taas delegoitu hallinta vaatii ohjelmiston suunnitteluvaiheessa suurempaa työpanosta. Lisäksi delegoidussa mallissa joudutaan miettimään, tarjotaanko käyttäjätodennukseen vaadittava hakemistopalvelu palveluntarjoajan toimesta vai käytetäänkö hajautetusti asiakkaiden omia hakemistopalveluita federaation kautta.

3.3.5 Tietomalli

Sovellustasolla moniasiakkuusympäristössä tärkeimmäksi osakysymykseksi muodostuu, miten SaaS-palvelun tarjoaja huolehtii tenanttien eli asiakkuuksien tietojen varastoinnista ja eristämistä siten, että tietovuotoja ei tapahdu asiakkuuksien välillä (Chong, Carraro, & Wolter, 2006). Tämä saattaa muodostua kynnyskysymykseksi, kun yritys harkitsee SaaS-palveluun siirtymistä paikallisesta omasta ympäristöstä, ja multitenantin tietoarkkitehtuurin suunnittelussa tuleekin panostaa tietovaraston rakenteeseen edellä mainittu näkökulma huomioiden (Chong et al., 2006).

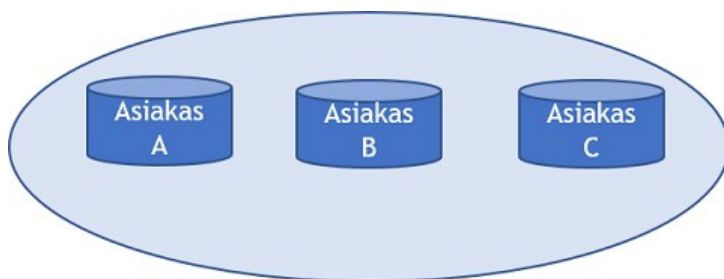
Moniasiakkuuskäyttöön suunnitellun sovelluksen tietokanta-arkkitehtuurin rakenne voidaan ajatella janalle, jonka ääripäissä ovat asiakkuuksien tiedon täydellinen eristäminen toisistaan asiakkuuskohtaisiin tietokantoihin ja toisessa päässä taas jaettu tietokantarakenne. Optimaalinen rakenne haetaan suunnittelussa jollakin tasolla janan keskeltä (Kuva 10), jossa vasen pää edustaa perinteisempää ja myös kustannuksiltaan suurempaa ratkaisua, kun taas oikeassa reunassa luodaan täysin jaettu tietokantaratkaisu, jolloin myös käyttöoikeuksien hallinnan tärkeys korostuu entisestään.



Kuva 10: Tietokannan rakenne multitenantissa ympäristössä (Chong et al., 2006).

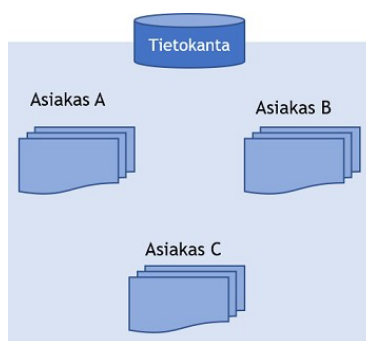
Chong et al. (2006) esittävät edellä kuvatun perusteella kolme perusmallia tietokanta-arkkitehtuurin suunnittelussa moniasiakkuusympäristössä toteutettavaksi.

Täysin *erillinen tietokanta* (*separate database*) luodaan jokaiselle tenantille (Kuva 11), jolloin laskenta-teho ja sovellus itsessään ovat käytössä jaetusti, mutta jokaisen asiakkuuden tietovarasto on loogisesti eristetty toisistaan (Chong et al., 2006). Tässä mallissa asiakkuuskohtaiset räätälöinnit ovat helpoiten toteutettavissa, koska erillinen tietokanta voidaan muuttaa rakenteellisesti ilman, että muutokset vaikuttavat muihin asiakkuuksiin. Varmistusten palauttaminen tietokantasolla vaikuttaa saatavuuteen ainoastaan asiakkuuskohtaisesti, mutta erillisten tietokantojen varmistaminen ja ylläpitäminen pitkällä aikavälillä lisää ylläpidollisia kustannuksia lisääntyvän tallennustilan ja laskentatehon kasvavina vaatimuksina (Chong et al., 2006).



Kuva 11: Erilliset tietokannat (Chong et al., 2006).

Seuraava taso matkalla jaettuun ympäristöön on malli, jossa asiakkuuksilla on *erilliset skeemat jaetussa tietokannassa* (*shared database, separate schema*). Tässä mallissa (Kuva 12) tenantin perustamisvaiheessa (uuden asiakkuuden perustaminen) luodaan ennalta määritellyn prosessin mukaisesti asiakkaalle oma skeema sekä tietokantataulut, jotka liitetään asiakkaan skeemaan (Chong et al., 2006). Lopuksi asiakkuuden käyttäjäryhmille delegoidaan käyttöoikeus kyseiseen kokonaisuuteen, minkä johdosta tietokantakyselyitä tehtäessä kukin asiakkuus saa saman tietokantahaun tuloksena vain oman käyttöoikeuden piiriin kuuluvaa tietoa. Asiakkuus siis on jaetusta tietokannasta huolimatta eristetty omaan hiekkalaatikkoonsa, jonka sisältä ei saada näkyvyyttä muiden asiakkuuksien vastaavaan kokonaisuuteen. Asiakkuuden ja siihen liittyvien peruskomponenttien luomisen jälkeen erillisen skeeman ansiosta tietokannan rakennetta voidaan räätälöidä asiakkaan haluamalla tavalla ilman, että muutokset näkyvät muille tenanteille.



Kuva 12: Jaettu tietokanta, erilliset skeemat (Chong et al., 2006).

Mallin käyttöönotto muuttuu suunnitteluvaiheessa haastavammaksi, jossa asiakkuuden perustaminen tulee mallintaa tarkasti ja automatisoida alusta loppuun saakka, sillä tässä vaiheessa tehdyt suunnitteluvirheet kertautuvat myöhemmin kaikille asiakkuuksille. Lisäksi ylläpidollisesta näkökulmasta tiedon palautus tietokannan varmistuksesta ei onnistu suoraan asiakkuuskohtaisesti, koska tietokannoista tehtävät varmistukset ovat tietokantakohtaisia. Näin ollen vikatilanteesta toipuminen kestää pidempään ja asiakkuuskohtainen tieto joudutaan poimaan tuotantoympäristön rinnalle palautetusta tietokannasta.

Jaetuimmassa mallissa kaikilla asiakkuuksilla on *yhteinen tietokanta, jossa käytetään yhteistä skeemaa (shared database, shared schema)*. Tällöin jokaisen tietokantataulun jokaisessa tietueessa kulkee mukana aina tenantin yksilöivä tunnus (Kuva 13), jonka avulla kaikissa kyse-lyissä suodatetaan asiakkuuskohtainen tieto. Malli on toimivin vaihtoehto silloin, kun halutaan maksimoida yhdestä instanssista palveltavien asiakkuuksien määrä (Chong et al., 2006) ja sama- aikaisesti pyritään optimoimaan alustan resurssien käyttö mahdollisimman tehokkaaksi.

AsiakasNRO	Asiakas	Osoite
465	AsiakasNRO	TuoteNRO TuoteNimi
2346	AsiakasNRO	Tilausnro TilausPVM
3452	46903	345348 2017-08-04
2434	23850	234290 2017-06-02
24	34890	304900 2017-04-03
	24290	220002 2017-02-01

Kuva 13: Jaettu tietokanta, jaettu skeema (Chong et al., 2006).

Tämän täysin jaetun mallin käyttöönotossa suunnittelun rooli korostuu etenkin tietoturvan alueella entisestään (Chong et al., 2006), eikä sitä voi painottaa liikaa. Asiakkuuskohtaisen tiedon palauttaminen vikatilanteessa muuttuu entistä haastavammaksi, koska asiakkuuksien tieto on keskitetty samoihin tietokantatauluihin ja näin ollen yhden asiakkuuden ongelman käsittely voi näkyä muille asiakkuuksille suorituskyvyn laskemisen ja pahimmassa tapauksessa virheellisten toimenpiteiden kautta sovelluksen käyttöasteen laskemisena.

3.4 Yritysmobiilihallinta

Mobiililaittehallinnalla (engl. Mobile Device Management) voidaan valvoa, suojata ja hallita käytössä olevia laitteita keskitetysti koko niiden elinkaaren ajan. Yritystasolla tämä tarkoittaa yleensä julkisen internetin yli tapahtuvaa sovellusjakelua, asetusten hallintaa yrityksen hallinnan piiriin kuuluvien puhelimien, tablettien ja muiden mobiililaitteiksi luokiteltavien laitteiden osalta (Johnson, 2011). Laittehallinnan tavoitteena yritystasolla on optimoida ja vakioi- da käyttöympäristö mahdollisimman tehokkaasti, millä pyritään saavuttamaan mahdollisim- man turvallinen ja kustannustehokas ympäristö liiketoiminnan käyttöön koko laite-elinkaaren ajaksi.

Alun perin ja ensisijaisesti MDM-hallinnalla siis on pyritty ottamaan yrityksen laitteet hallin- taan ja pakottamaan tiettyjä asetuksia ja yleensä rajoituksia yrityksen omistamiin laitteisiin. Viime vuosina kuitenkin käytössä olevien päätelaitteiden hankinta on muuttunut vähitellen kuluttajalähtöisemmäksi (engl. consumerization of IT), mikä käytännössä tarkoittaa sitä, että joissakin yrityksissä käyttäjät voivat itse hankkia tietyissä rajoissa haluamansa laitteen. Täl-

löin haasteeksi syntyy tietohallinnon kannalta yrityksen tiedon ja käyttäjän yksityisen tiedon turvallinen eristäminen toisistaan. Tältä pohjalta on syntynyt käsite yritysmobiilihallinta (engl. Enterprise Mobility Management = EMM).

Citrixin näkemyksen mukaan (Citrix Techtarget, 2012) yritysmobiilihallinnan voidaan katsoa käsittävän varsinaisen laitehallinnan ja myös laitteissa käytettävien sovellusten, dokumenttien ja sähköpostin sisällön hallinnan yhtä lailla. Tällä saavutetaan tarkasti määritellyn tietoturvan lisäksi myös joustava käyttökokemus käyttäjille siten, että yrityksen tiedoksi määritelly data voidaan etähallitusti pyyhkiä päätelaitteesta ilman, että esimerkiksi yksityinen sähköpostiprofiili poistuu.

Yritysmobiilihallinnan (EMM) voidaan katsoa sisältävän seuraavat osa-alueet (Pierer, 2016):

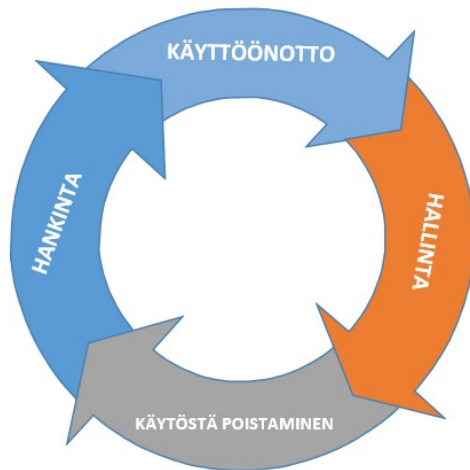
- Mobiililaitehallinta (MDM=Mobile Device Management)
- Mobiilisovellushallinta (MAM=Mobile Application Management)
- Mobiilitietoturvan hallinta (MSM=Mobile Security Management)

3.5 Työasemien elinkaarenhallinta

Elinkaarimalli käsittää vaiheet aina laitehankinnasta asianmukaiseen käytöstä poistamiseen asti, johon sisältyvät prosessit, tehtävät ja resurssit tarvittavien toimenpiteiden suorittamiseksi, jotta laitteet pysyvät toimintakuntoisina ja käyttökelpoisina liiketoimintaympäristössä. Elinkaaren hallinta voidaan tuottaa yhtiön omilla resursseilla tai vaihtoehtoisesti ulkoistaa osittain tai kokonaan tällaista palvelua tuottavalle kumppanille (Gartner, 2006).

Elinkaarimallin käyttöönotto edellyttää realistista vaatimusten ja tavoitteiden arviointia koko organisaation ja toisaalta toimitusketjun läpi, jotta toiminta saadaan optimaalisesti kustannustehokkaaksi. Tavoitteena on näin ollen suoraviivainen prosessi, missä ei tehdä samoja päällekkäisiä toimenpiteitä turhaan ja useiden resurssien toimesta. Puutteellinen suunnittelu johtaa helposti lopputulokseen, joka ei paranna kustannusrakennetta toivotulla tavalla.

Elinkaarimalli esitetään tyypillisesti kiertokulkumallina (Kuva 14: Elinkaarimalli.Kuva 14), missä kuvatut päävaiheet sisältävät useita alitehtäviä ja näiden sisällöt sekä tarkempi luokittelu ovat organisaatiokohtaisia omien tarpeiden mukaisesti muokattuja. Tyypillinen mittari toimivasta elinkaariprosessista on esimerkiksi se, että vuosittain käytöstä poistetut laitteet todellakin poistuvat yrityksen varastohuoneista ja ne hävitetään tai kierrätetään asianmukaisesti ja tietoturvasyistä.



Kuva 14: Elinkaarimalli.

Elinkaaren hallinnassa hankintaprosessin taustalla on valittu tietyillä kriteereillä laitetoimittajat, joilta tilataan sovitun hankintakanavan välityksellä halutut laitekoonpanot. Perinteisesti suuret yritykset ovat sisällyttäneet hankintaan myös käyttöönottovaiheeseen kuuluvan asiakkaan määrittelemän esiasennuksen. Esiasennus sisältää käyttöjärjestelmän ja valittujen sovellusten asentamisen käyttöönotettavaan laitteeseen. Valmiiksi asennettu laite toimitetaan asiakkaan määräämään osoitteeseen, missä mahdollisesti lähituki tai loppukäyttäjä omatoimisesti ottaa laitteen käyttöön.

Työasemien elinkaarimallissa pääalueiden sisältö voidaan esittää seuraavasti (Gartner, 2006):

1. Hankinta

- vaatimusmäärittely
- laitetoimittajan valinta ja hallinta.
- sopimushallinta.

2. Käyttöönotto

- laiteasetusten määrittely ja käyttöönotto.
- käyttöjärjestelmän ja sovellusten esiasennus.
- migraatio (vanha laite korvataan uudella).
- toimitus loppukäyttäjälle.

3. Hallinta

- service desk- ja lähitukipalvelut.
- koulutus.
- päivitysten jakelu.

- muuttuneiden asetusten jakelu.
 - varmistukset
4. Käytöstä poistaminen
- laitteen poistaminen kirjanpidosta.
 - laitteen tietoturvallinen tyhjennys.
 - lisenssien vapauttaminen.
 - laitteen asianmukainen kierrätys.

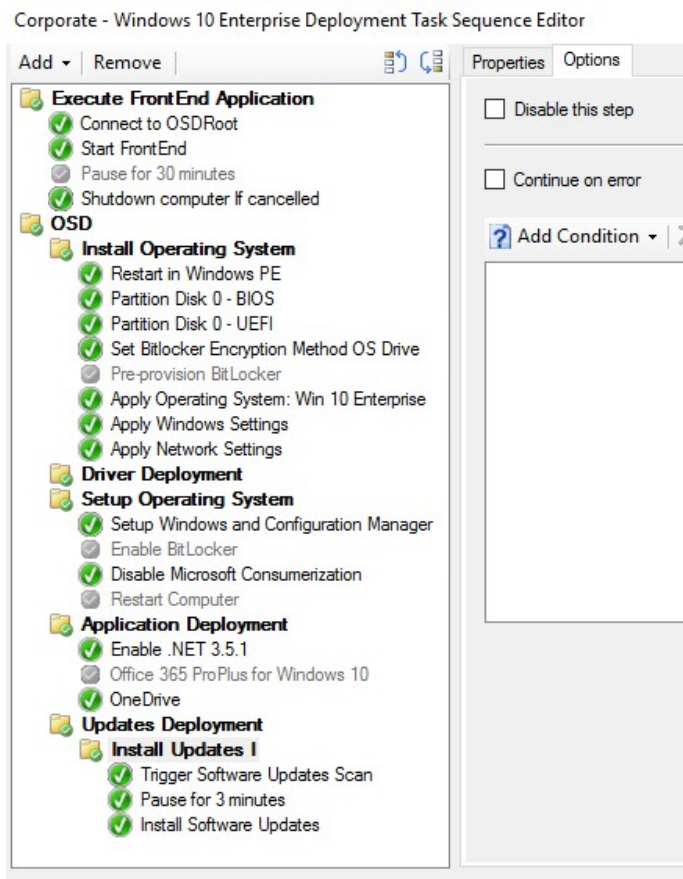
3.6 System Center Configuration Manager

Microsoft System Center Configuration Manager on System Center -tuoteperheeseen kuuluva hallintatuote, jolla voidaan hallita työasemien elinkaari esiasennuksen, asetusten hallinnan, päivitysten ja raportoinnin osalta (Microsoft Corporation, 2017e). Configuration Manager (SCCM) on suunniteltu yrityskohtaiseksi sisäverkossa käytettäväksi hallintatuotteeksi ja lisätoiminteilla on mahdollista rakentaa työasemille suojattu, SSL-salattu, yhteys SCCM:n hallinta- ja jakelupisteisiin. Lisäksi muut päätelaitteet, kuten puhelimet ja tabletit, voidaan liittää SCCM:n hallinnan piiriin Intune-integraation kautta, jolloin päätelaitteiden rajapintana käytetään Microsoftin pilvipalvelua.

Microsoftin strategia on nykyisin ”Cloud First” eli tavoitteena on suunnata voimakkaasti pilvipalveluihin, mutta samanaikaisesti Configuration Manager -tuotetta kehitetään jatkuvasti vastaamaan uusia tarpeita ja tuotteesta julkaistaankin tällä hetkellä uusi versio kaksi kertaa vuodessa. Merkitsevää on, että SCCM:ään tuodaan koko ajan liittymiä Microsoftin Azure-pilvipalvelussa oleviin palveluihin, kuten Upgrade Analytics ja Advanced Threat Protection.

3.6.1 Esiasennusominaisuudet

SCCM:n vahvuutena on mahdollisuus toteuttaa työasemien esiasennus alusta lähtien halutulla kokoonpanolla käyttäen hallittua asennusjärjestystä (engl. Task Sequence). Esiasennukseen sisältyy tyypillisesti kiintolevyn alustus, usein kiintolevyn salaus, käyttöjärjestelmän perusasennus ja mallikohtainen laiteajuriasennus. Lopuksi asennetaan tarkoituksen mukaiset liike-toiminnassa tarvittavat sovellukset sekä tarvittavat räätälöinnit, kuten haluttujen Windows-komponenttien poistaminen tai lisääminen (Kuva 15).



Kuva 15: SCCM:n esiasennusprosessi (Savolainen, 2017c).

3.6.2 Sovellushallintaominaisuudet

Sovellusjakelu on alusta lähtien ollut SCCM:n ja sen edeltäjän alkuperäinen päätoiminnallisuus ja versiosta 2012 lähtien tuotteeseen on sisällytetty erityinen sovellushallintamalli (engl. application deployment model), joka käsittää hyvin laajan valikoiman erilaisia työasemiin ja mobiililaitteisiin jaeltavia sovelluspakettityyppejä (Bennett et al., 2016). Lisäksi jaeltaviin sovelluksiin voidaan liittää monipuolisesti riippuvuuksia (engl. dependence), jolloin voidaan mallintaa ja automatisoida tarvittavien esivaatimusten asentaminen sekä korvaavuus (engl. supersedence), jonka avulla voidaan tuotantoympäristössä korvata nykyinen sovellus toisella vastaavalla (Bennett et al., 2016).

Julkaistava sovellus on kokonaisuus, joka sisältää sovelluksen perustiedot, kuten valmistaja, nimi, versio jne. Itse sovellusobjektin sisällä voidaan määritellä useita julkaisutyppejä (engl. deployment type), jotka voidaan suodattaa hyvin monipuolisilla säännöillä esimerkiksi erikseen tiettyyn käyttöjärjestelmäversioon (Krstevski, 2014). SCCM osaa käsitellä ja raportoida työasemien palauttavat tilatiedot hallitusti näiden sääntöjen perusteella.

Jokaiseen Windows-työasemille jaeltavaan sovelluksen julkaisutyyppiin voidaan liittää poistokomento (engl. uninstall), jota hyödynnetään tarvittaessa esimerkiksi uuden version jakelussa tai korvaamistoimenpiteessä automaattisesti. On huomattava, että sovellusten asennus- ja poistokomennot tulee etsiä jakelutyövälineestä riippumatta aina sovelluksen valmistajan tai muiden tietolähteiden avulla. Esimerkiksi Java 8 -sovelluksen (Kuva 16) vastaavat komennot.

Specify the command used to install this content.

Installation program:

Installation start in:

Configuration Manager can remove installations of this content if an uninstall program is specified

Uninstall program:

Uninstall start in:

Kuva 16: Java 8 -sovelluksen asennus- ja poistokomennot (Savolainen, 2017c).

Kaiken kaikkiaan sovellusjulkaisutyyppejä on mahdollista luoda kaikille tuetuille käyttöjärjestelmille kuvan mukaisesti. Näistä Windows 10 -työasemille jaeltavia tyyppejä ovat:

- *Windows Installer (*.msi file)*, joka on perinteinen Microsoftin luomaan standardiin perustuva sovelluspaketti.
- *Script installer*, joka käsittää yleisesti kaikki komentoriviltä suoritettavat asennuskomennot tai muut työasemille jaeltavat komennot.
- *Windows app package (*.appx)*, joka on Windowsin modernin sovelluksen erikseen ladattu asennustiedosto.
- *Windows app package (in the Windows Store)*, joka edellyttää integraation perustamista SCCM-hallintaympäristön ja Microsoft Store for Business -portaalin välille.

Windows Installer (*.msi file)
Windows Installer (*.msi file)
Windows app package (*.appx, *.appxbundle)
Windows app package (in the Windows Store)
Script Installer
Microsoft Application Virtualization 4
Microsoft Application Virtualization 5
Windows Phone app package (*.xap file)
Windows Phone app package (in the Windows Phone Store)
App Package for iOS (*.ipa file)
App Package for iOS from App Store
App Package for Android (*.apk file)
App Package for Android on Google Play
Mac OS X
Web Application
Windows Installer through MDM (*.msi)

Kuva 17: SCCM:n sovellusjakelutyytit (Savolainen, 2017c).

3.6.3 Päivitysten hallintaominaisuudet

Päivitysten hallintaominaisuudet ovat parantuneet jokaisen Configuration Manager -version myötä ja nykyisin kaikki päivitystenhallintaan liittyvät toimenpiteet voidaan tehdä puhtaasti SCCM:n toiminnollisuuksiin perustuen. Kuukausittaiset päivitykset voidaan tuottaa hallittavaan ympäristöön täysin manuaalisesti, täysin automatisoidusti tai valitsemalla toimintamallin näiden ääripäiden keskeltä (Bennett et al., 2016). Rajapintana hallittavien työasemien ja Configuration Manager -ympäristön välillä toimii Software Update Point (SUP) -roolin omaava palvelin ja SCCM:n versiosta 1702 (Microsoft Corporation, 2017g) alkaen suurten ympäristöjen verkkokuormaa voidaan tasata hallitusti useille SUP-palvelimille sijaintiin perustuen.

Windows 10 -käyttöjärjestelmän julkaisun jälkeen Microsoft on tarkentanut päivityspolitiikkaa siten, että jatkossa käyttöjärjestelmästä julkaistaan uusi versio kaksi kertaa vuodessa ja nämä versiot vanhenet tietyssä aikaikkunassa, jolloin tietoturva- ja korjauspäivityksiä ole saatavilla kyseiselle versiolle (Microsoft Corporation, 2017d). Näin ollen yritykset pakotetaan päivittämään käyttöjärjestelmiä säännöllisesti, mikä asettaa yritysten tietohallinto-osastot uuteen tilanteeseen ja luomaan toimivat testaus- ja päivitysprosessit.

SCCM tukee suoraan näiden uusien Windows-versioiden hallittua jakelua tuotantoympäristössä, mikä on oleellinen ominaisuus vertailtaessa esimerkiksi Intunen MDM-hallinnan ja SCCM:n hallintaominaisuuksia keskenään.

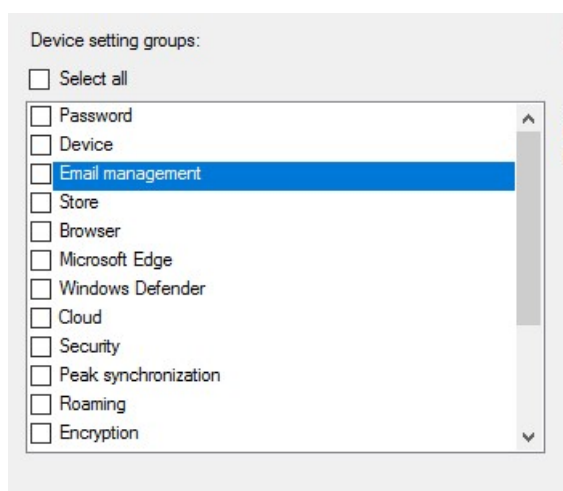
3.6.4 Raportointiominaisuudet

Configuration Manager sisältää raportointiominaisuuden, joka pohjautuu puhtaasti SQL Reporting Services -palveluun ja mahdollisuudet luoda hyvin monimutkaisia ja räätälöityjä raportteja ovat rajattomat. SCCM-tuotteen mukana on valmiina lähes 500 erilaista raporttia Microsoftin tuottamana, jotka ovat välittömästi käyttöönoton jälkeen hyödynnettävissä (Bennett et al., 2016).

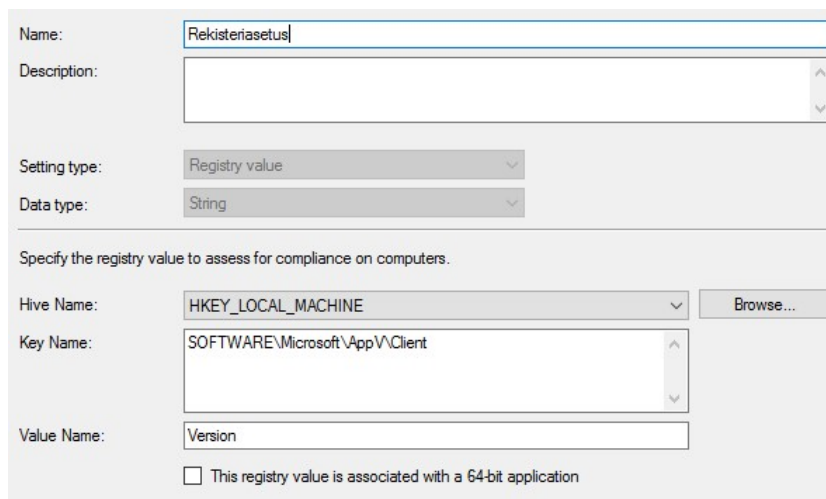
Raportit voidaan tuottaa saataville täysin teknisen SCCM-hallinnan ulkopuolelle siten, että käyttöoikeuksiin perustuen käyttäjätunnuksilla on mahdollisuus tulostaa raportteja suoraan selainpohjaisesti. Lisäksi voidaan tuottaa automaattisesti määräajoin raportteja suoraan ennalta määrättyihin sähköpostiosoitteisiin (Krstevski, 2014).

3.6.5 Asetusten hallinta

Asetusten hallinnassa on keskeistä mahdollisuus rajata päätelaitteen käyttömahdollisuuksia toisaalta käytettävyyden yksinkertaistamiseksi ja toisaalta myös vahinkojen estämiseksi (Krstevski, 2014). Configuration Managerilla on mahdollista jakaa kaikille päätelaitteille hyvin monipuolisesti asetuksia ja erilaisten toimintojen rajoituksia tai ohjauksia hyödyntäen valmiita velhopohjaisia perusasetuksia (Kuva 18) sekä lisäksi luomalla erittäin monimutkaisia räätälöityjä asetuskokonaisuuksia.



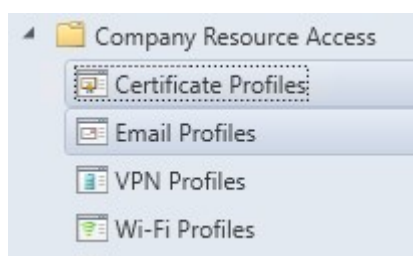
Kuva 18: Esimerkki velhopohjaisesta asetuksesta (Savolainen, 2017c).



Kuva 19: Esimerkki räätälöitävästä konfiguraatiosta (Savolainen, 2017c).

Configuration Managerilla voidaan jakaa sertifikaatti-, sähköposti-, VPN- ja WiFi-profiileja Windows-työasemille tarpeen mukaan (Kuva 20). Tuetut sertifikaattityypit (SCEP, PFX ja CER) käsittävät kaikki Windows-ympäristössä yleisimmin käytössä olevat sertifikaatit ja näin pystytään automatisoidusti luomaan tarvittava pohja pääsyyte yrityksen toimintaympäristöön. Li-

säksi voidaan automaattisesti tuoda sähköpostisovelluksille, tietyille VPN-ratkaisuille ja langottomalle lähiverkkoyhteydelle tarvittavat asetusprofiilit.

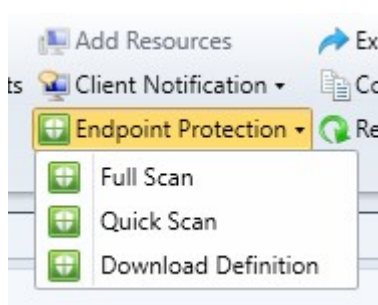


Kuva 20: SCCM:llä jaeltavat profiilityypit yrityksen toimintaympäristöön pääsyä varten (Savolainen, 2017c).

3.6.6 Etähallintaominaisuudet

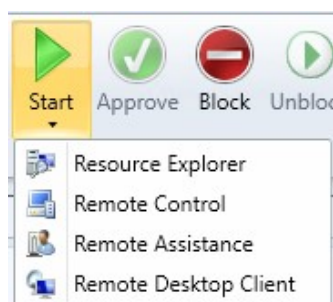
Etähallintaominaisuuksilla tarkoitetaan hallintakonsolilta käsin aloitettavaa toimenpidettä kohdetyöaseman suuntaan, ja teknisesti osa näistä toimenpiteistä edellyttää suoraa lähiverkkoyhteyttä työasemalle palomuurisäännöt huomioiden. Toinen osa puolestaan ei edellytä suoraan yhteyttä vaan etäkomento siirtyy työasemalle suoritettavaksi siinä vaiheessa, kun kyseinen laite ottaa seuraavan kerran yhteyttä hallintaympäristöön.

Configuration Manager sisältää virustorjuntakomponentin (EndPoint Protection), joka on integroitu SCCM-hallintaan ja kyseistä sovellusta voidaan ohjata etäkomennoina määräämällä välittömän virusskannauksen tai virustietokannan päivityksen (Kuva 21) halutulle laitekokoelmalle.



Kuva 21: EndPoint Protection -sovelluksen tukemat etähallintakomennot (Savolainen, 2017c).

Etätukiyhteys (engl. Remote assistance) on mahdollista muodostaa suoraan SCCM:n hallintakonsolista (Kuva 22) työasemalle silloin, kun työasemaan on suora pääsy lähiverkkoyhteydellä. Teknisesti etätuessa hyödynnetään Windows-käyttöjärjestelmän omaa etätukiyhteysoinaisuutta. Muualla on sisäverkossa sijaitsevaan työasemaan ei saada etätukiyhteyttä näillä ominaisuuksilla vaan tähän on hyödynnettävä kolmannen osapuolen vastaavia työvälinesovelluksia (Microsoft Corporation, 2017e).



Kuva 22: SCCM:n etätukiyhteys (Savolainen, 2017c).

3.7 Microsoft Intune

3.7.1 Yleisesti

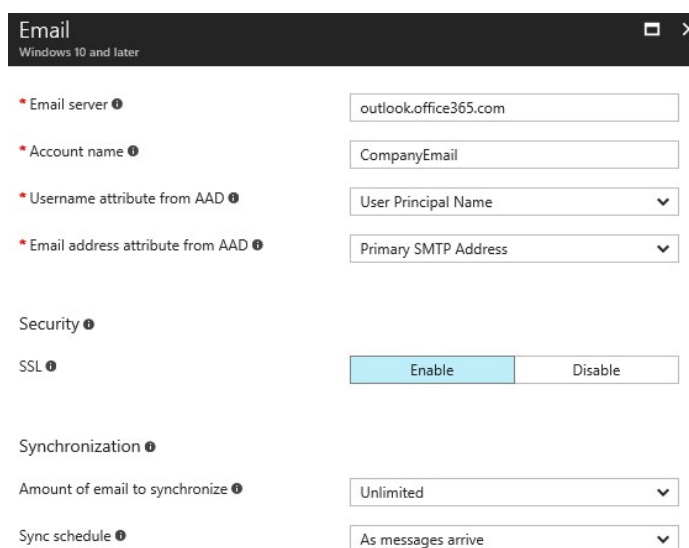
Microsoft Intune on multitenantti, pilvipohjainen mobiilihallintajärjestelmä, joka on alun perin suunnattu pienille ja keskisuurille yrityksille. Käytännössä teknisen 50 000 päätelaitteen ylärajan poistuttua (Kaskisaari, 2016) hallintaympäristö voi periaatteessa soveltua erittäin suurien liiketoimintaympäristöjen ylläpitoon. Intune kuuluu osana laajempaan EMS (Enterprise Mobility+Security) -kokonaisuutta ja Intunen roolina on nimenomaan erilaisten päätelaitteiden hallinta. (Microsoft Corporation, 2017b)

Intune tukee, Windows-, Windows Mobile-, IOS-, Android- ja Mac OSX -käyttöjärjestelmillä varustettuja päätelaitteita, joiden hallittavissa olevat ominaisuudet riippuvat toisaalta Intunessa julkaistuista ominaisuuksista ja toisaalta taas päätelaitteen valmistajan avaamista rajapinnoista yksittäisiä hallintatoimenpiteitä ajatellen. Tavallisimpia päätelaitteisiin kohdistuvia hallintatoimintoja ovat esimerkiksi PIN-koodin pakottaminen, sovellusten jakelu, sähköpostiprofiilien asettaminen sekä laitteen etätyhjennys.

Intunen käyttöön ottaminen on helppoa ja tyypillisesti multitenantille pilvipalvelulle alusta on määrämuotoinen kaikille asiakkaille, eikä perinteistä palvelimien tai sovellusten asentamista edellytetä. Toki peruskäyttöönotto ei vielä käytännössä tuo varsinaista hyötyä hallintatuotteesta, mikäli asetuksia ja sovelluksia ei konfiguroida halutulla tavalla. Käyttöönottoa ennen tulee kuitenkin päättää, halutaanko Intunea käyttöön erillisenä tuotteena vai halutaanko se integroida Microsoftin System Center Configuration Manager -tuotteen kanssa, jolloin puhutaan niin kutsutusta hybridiratkaisusta.

Laitteiden hallintaan liittäminen edellyttää osalta päätelaitetyypeistä erillistä verkkokaupasta ladattavaa ilmaista sovellusta, joka toimii rajapintana Intunen ja laitteen välillä. Perusideana on, että loppukäyttäjät liittävätkin laitteensa itsepalveluna hallinnan piiriin ajasta ja paikasta riippumattomasti ilman tietohallintopalveluita.

Intune-hallinnassa tyypillisiä perustoimintoja ovat muun muassa PIN-koodin asettaminen sekä sähköpostiprofiilin automaattinen jakelu (Kuva 23). Lisäksi voidaan laitetyypistä riippuen asettaa erilaisia rajoituksia, kuten esimerkiksi kameran ja bluetooth-yhteyden käytön estäminen. Turvallisuuden kannalta mielenkiintoinen uudehko ominaisuus on ehdollinen pääsynhallinta (engl. conditional access), jolla voidaan määritellä, milloin laitteella on pääsy esimerkiksi yrityksen sähköpostiin. Yleisesti ensisijainen tarkistus on, että laitteessa tulee olla PIN-koodi asetettuna, mikä suojaa varsin tehokkaasti kadonneen puhelimen tiedot.



Email
Windows 10 and later

- * Email server outlook.office365.com
- * Account name CompanyEmail
- * Username attribute from AAD User Principal Name
- * Email address attribute from AAD Primary SMTP Address

Security

SSL Enable Disable

Synchronization

Amount of email to synchronize Unlimited

Sync schedule As messages arrive

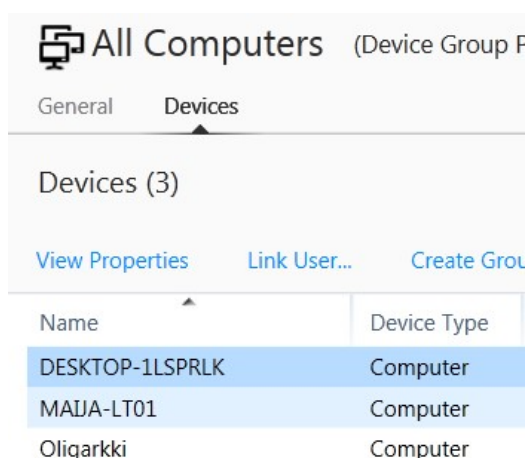
Kuva 23: Sähköpostiprofiilin asettaminen Intune-hallinnassa (Savolainen, 2017b).

Intunen hallinta uudistui keväällä 2017, kun Microsoft yhdisti alun perin erillisen hallintaympäristön osaksi kokonaisvaltaista Azure-pilvipalveluiden hallintaympäristöä. Tämä on looginen jatkumo sille, että monet Intune-hallinnassa hyödynnettävät ominaisuudet, kuten käyttäjäryhmät, käyttöoikeudet ja pääsynhallinta, perustuvat Azuressa jo aiemmin hallittuihin osaluokiin.

3.7.2 Työasemalaitteiden hallinta ja Windows 10

Microsoft Intunessa on alusta lähtien ollut mahdollisuus liittää hallintaan myös tavallisia PC-laitteita, alun perin erillisellä Intune Client -hallintaohjelmistolla, minkä avulla voidaan kerätä inventaariotietoa, jakaa sovelluksia sekä kuukausittaisia tietoturvapäivityksiä. Laittehallinnassa näin liitetyt työasemat näkyvät Computer-tyyppisinä laitteina (vrt. Mobile Device), mistä tiedetään kyseessä olevan (Kuva 24) nimenomaan normaali työasemalaite. Alun perin Intune-tuotteen ilmestyessä markkinoille tietokonelaitteiden hallinta olikin ylipäättään ainoa

mahdollinen hallintaominaisuus ja todellisten mobiililaitteiden hallinta oli vasta tulossa oleva ominaisuus.



Kuva 24: Työasemalaitteita vanhasta Intunen hallintanäkymässä (Savolainen, 2017b).

Windows 10 -käyttöjärjestelmän myötä Microsoft laajensi mobiililaitte käsitteen merkityksen käsittämään myös työasemalaitteita. Käytännössä Windows 10 -käyttöjärjestelmään on integroitu MDM-standardiin (Open Mobile Alliance, 2017) perustuva ohjelmistokomponentti, minkä välityksellä Windows-työasema voidaan liittää Intunen MDM-hallintaan.

Keväällä 2017 Microsoft siirsi Intunen toiminnallisuudet osaksi suurempaa Azure-palvelukokonaisuutta ja samanaikaisesti Intune-clientin käyttö on jäämässä asteittain pois käytöstä. Vanhan Intune-hallintaympäristön kautta sen käyttäminen on vielä mahdollista, mutta poistuvaan hallintatapaan ja sen tarjoamiin ominaisuuksiin ei voida enää tässä vaiheessa nojautua. Näin ollen ainoana vaihtoehtona on jatkossa pelkästään mobiilihallintaominaisuudet pilvipalveluiden käytössä.

3.7.3 Hallintaominaisuudet Intune Clientilla ja MDM-hallinnalla

Windows 10 -käyttöjärjestelmän ilmestyessä vuoden 2015 kesällä tuli aiemmin esitelty MDM-ominaisuus saataville. Tähän asti käytetty Intune Client -hallintasovellus oli edelleen käytävissä siten, että laite oli saman aikaisesti liitettyä uuden MDM-hallintarajapinnan välityksellä Intune-hallinnan piiriin. Vuoden 2016 alussa Microsoft julkaisi normaalisti uuden asennusversion Intune Client -sovellukselle ja tämän julkaisun myötä asennusohjelmaan lisättiin tarkistus, joka estää Intune Clientin asennuksen, jos laite on jo liitetty MDM-rajapinnan kautta Intune-hallinnan piiriin. Lopputulemana päädyttiin siis nykyiseen tilanteeseen, jossa joudutaan valitsemaan näiden kahden hallintavaihtoehdon välillä ja näin olleen valitsemaan hallittavien ominaisuuksien välillä.

Käytännön kannalta merkittävää tässä muutoksessa oli se, että kyseisestä muutoksesta ei koskaan tiedotettu virallisesti vaan asia tuli kaikille asiakkaille eteen yllätyksenä, mikä aiheutti varmasti paljon haasteita, koska oletusarvoinen tilanne muuttuikin yhtäkkiä. Samanaikaisesti Microsoft on vienyt voimakkaasti kehitystä MDM-hallintaan, josta kuitenkin puuttuu toistaiseksi merkittäviä ominaisuuksia työasemahallinnan kannalta.

Lopputulena on päädytty tilanteeseen, jossa vaihtoehtoinen Microsoftin tarjoamaista hallintamahdollisuuksista ovat MDM-hallinta Intunesta tarjottuna tai sisäverkkoon perustettava Configuration Manager-palvelu, joista jälkimmäinen tarjoaa tällä hetkellä käytännössä rajatut mahdollisuudet räätälöidä tarvittavat ominaisuudet hallittavaa ympäristöä silmällä pitäen. Tämä puolestaan johtaa valintatilanteeseen Intunen ja SCCM:n välillä.

4 Tutkimustulokset

Tässä tutkimuksessa on koostettu tutkijan asiakasympäristöissä hankkimaan omakohtaiseen kokemukseen perustuvaa tietoa Windowsin hallittavista ominaisuuksista ja edelleen näistä on koostettu vertailutaulukot, joista ilmenee, kummalla hallintatavalla kyseistä ominaisuutta voidaan ohjata. On huomattava kuitenkin, että vertailu tehdään sillä perusteella, miten kyseinen ominaisuus on hallittavissa suoraan valmiilla ominaisuuksilla eli tutkimuksen tässä osiossa ei oteta kantaa siihen, millaisilla räätälöintimahdollisuuksilla kyseinen ominaisuus voisi olla hallittavissa.

4.1 Vertailu hallintaominaisuuksista

Hallittavia ominaisuuksia Configuration Managerin (SCCM) ja Windows 10 MDM-liitoksen välillä verrattiin taulukoissa (taulukot 3-10), joissa ominaisuudet esitetään alkuperäisin termein englanniksi siten kuin ne Intune-hallinnassa näkyvät. Taulukoiden pohjana on käytetty Aaron Parkerin laatimia taulukoita (Parker, 2016) ja niitä on täydennetty ja tarkennettu tähän tutkimukseen soveltuvaksi.

Taulukko 3: Yleiset ominaisuudet.

Ominaisuus:	SCCM:	Intune MDM:
Automaattinen liittyminen laitehallintaan Azure AD -liitoksen kautta.	Ei	Kyllä
Yhteensopivuusasetusten hallinta	Ei	Kyllä
Endpoint Protection -tuotteen hallinta	Kyllä	Kyllä
Endpoint Protection -tuotteen raportointi	Kyllä	Ei
Lisenssien hallinta	Kyllä	Ei

Jakelupisteen sijoittaminen lähiverkkoon	Kyllä	Ei
--	-------	----

Taulukko 4: Esiasennusominaisuudet.

Ominaisuus:	SCCM:	Intune MDM:
Käynnistys Windows PE-tilaan	Kyllä	Ei
Hallittu asennusjärjestys (Task Sequence)	Kyllä	Ei

Taulukko 5: Sovellushallintaominaisuudet.

Ominaisuus:	SCCM:	Intune MDM:
Sovellusjakelu (yksittäinen MSI-tiedosto)	Kyllä	Kyllä
Sovellusjakelu (MSI+muita tiedostoja)	Kyllä	Ei
Sovellusjakelu (EXE-tiedosto + muita tiedostoja)	Kyllä	Ei
Modernien sovellusten jakelu	Kyllä	Kyllä
Integraatio Microsoft Store -sovelluskauppaan	Kyllä	Kyllä
Integraatio Apple Store -sovelluskauppaan	Kyllä	Kyllä
Integraatio Google Play for Business -sovelluskauppaan	Kyllä	Kyllä
Asennetun sovelluksen poistokomento	Kyllä	Ei
Sovellus tuominen käyttäjien saataville itsepalvelullisesti.	Kyllä	Kyllä

Taulukko 6: Päivitysten hallintaominaisuudet.

Ominaisuus:	SCCM:	Intune MDM:
Windows 10 versiopäivityksen jakelu	Kyllä	Ei
Päivitysten viivästäminen rajattomasti	Kyllä	Ei
Versiopäivitysten viivästäminen	Kyllä	Kyllä
Koontipäivitysten viivästäminen	Kyllä	Kyllä
Yksittäisten päivitysten hyväksyminen	Kyllä	Ei

Taulukko 7: Raportointiominaisuudet.

Ominaisuus:	SCCM:	Intune MDM:
Peruslaiteraportti	Kyllä	Kyllä
Perinteisten sovellusten perusraportti	Kyllä	Ei
Modernien sovellusten perusraportti	Kyllä	Kyllä
Omien raporttien luontimahdollisuus	Kyllä	Ei
SQL Reporting Services -ominaisuudet	Kyllä	Ei

Taulukko 8: Asetusten hallinta (engl. configuration policy).

Ominaisuus:	SCCM:	Intune MDM:
Yleiset käytön rajoitukset	Kyllä	Kyllä

Sähköpostiprofiilin hallinta	Kyllä	Kyllä
PKCS-sertifikaattien jakelu	Kyllä	Kyllä
SCEP-sertifikaattien jakelu	Kyllä	Kyllä
PKCS-sertifikaattien jakelu	Kyllä	
VPN-profiilien jakelu	Kyllä	Kyllä
Windows Defender ATP	Kyllä	Kyllä
OMA-URI -asetusten jakelu	Kyllä	Kyllä
WiFi-profiilien jakelu	Kyllä	Kyllä

Taulukko 9: Etähallinta- ja etätukitoimet (engl. remote tasks ja remote assistance).

Ominaisuus:	SCCM:	Intune MDM:
Täydellinen haittaohjelmien skannaus	Kyllä	Ei
Nopea haittaohjelmien skannaus	Kyllä	Ei
Virustietokannan päivitys	Kyllä	Ei
Uudelleenkäynnistys	Ei	Ei
Asetusten päivitys	Ei	Ei
Inventaarion päivitys	Ei	Ei
Etälukitus	Ei	Kyllä
Fresh Start -ominaisuus	Ei	Kyllä
Sisäänrakennettu etätuki	Kyllä	Ei

4.2 Tapaustutkimuksen pohjalta havaitut kehityskohteet

Tutkimuksen aikana havaittiin yhdistettynä käytännön kokemuksiin asiakasympäristöissä kaksi merkittävää puutetta, jotka hidastavat tai estävät yrityksen siirtymistä työasemien osalta puhtaasti pilvipalveluiden käyttöön nykyisten sisäverkossa käytettävien yrityskohtaisten hallintavälineiden sijaan. Microsoftin tuotteista yrityskohtaisesti käyttöönotettava tuote on Microsoft System Center Configuration Manager (Microsoft Corporation, 2017e), jolla pystytään tällä hetkellä vastaamaan käytännössä kaikki mahdollisiin tarpeisiin erilaisten päätelaitteiden elinkaaren hallinnassa.

Yleisenä haasteena on jo aiemmin todettu se, että Intune Client ja MDM-rajapinta tarjoavat toiminnallisuuksia siten, että ne eivät ole pääsääntöisesti saatavilla molemmilla hallintatavoilla. Samanaikaisesti Microsoft niin ikään ei tue molempien hallintatapojen käyttämistä yhtäaikaaisesti, mikä on jo itsessään merkittävä este Intune-tuotteen käyttöönottamiseen työasemahallinnassa. Asiakasympäristöissä saatujen kokemusten sekä useiden pre-sales -tapaamisten yhteydessä merkittävimmit yksittäisiksi ominaisuuksiksi ovat nousseet seuraavassa kuvatut ongelmakohdat.

Perinteinen esiasennuksen (engl. pre-installation) mahdollisuus puuttuu kokonaan, jolloin käyttäjillä olisi laitteen saapuessa kaikki sovellukset yms. tarpeellinen asennettu valmiiksi ja käyttäjän tai ICT-henkilöstön toimia ei tarvita. Pilvipalvelumallissa sovellukset voidaan jakaa työasemille jälkikäteen ja viiveellä, mikä aiheuttaa sen, että käyttäjä ei tiedä, milloin työasema on käyttövalmis ja toisaalta, että laitetta ei esimerkiksi sammuteta kesken sovellusasennuksen. Näin toimittuna siis perinteinen hallittu käyttöönotto muuttuu itsepalvelulliseksi käyttöönotoksi, mikä sisältää vikaantumisriskejä ja näin kasvattaa kustannuksia ja työmäärää.

Sovellusjakelu MDM-hallinnan kautta on ominaisuuksiltaan rajoitettu siten, että työasemaympäristöön voidaan julkaista ainoastaan sovelluksia, jotka on paketoitu yhteen MSI-pakettiin. Vertailutaulukosta (taulukko 2) ilmenee kolmesta mahdollisesta sovelluspakettiyhdistelmästä ainoastaan yksi eli Software Deployment via Single MSI on käytettävissä MDM-hallinnan kautta, kun taas kaikki kolme vaihtoehtoa ovat käytettävissä Intune Clientia käyttämällä. Käytännössä valtaosa sovelluksista on paketoitu siten, että asennusmedia koostuu useammasta kuin yhdestä tiedostosta tai asennusmedia on kokonaan muussa tiedostomuodossa, tyypillisesti EXE-muotoisena. Tästä seuraa se, että suurimmalle osalle julkaistavista sovelluksista joudutaan tuottamaan erillinen asennuspaketti tähän erikoistuneelta palvelutoimittajalta.

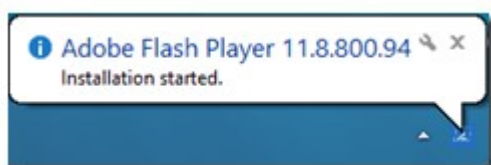
Sovellusinventaario (engl. software inventory) on käytettävissä kummallakin vaihtoehdolla, mutta MDM-hallinnan välityksellä saadaan työasemilta inventaario ainoastaan moderneista sovelluksista (Microsoft Corporation, 2017f). Käytännössä tuotantosovellukset ovat edelleen perinteisiä niin kutsuttuja legacy-sovelluksia, joiden käytössä olevista määristä ja versioita yritysten tietohallinnot haluavat kerätä tietoa muun muassa lisenssinhallintaa silmällä pitäen. Käytännön esimerkkejä tällaisista sovelluksista ovat muun muassa Microsoft Office, Microsoft Visio, Microsoft Project, Adobe Acrobat, Mozilla Firefox jne. Tällä hetkellä täydellinen sovellusinventaario on mahdollista ainoastaan Intune Clientin välityksellä.

4.3 Provisiointisovellus

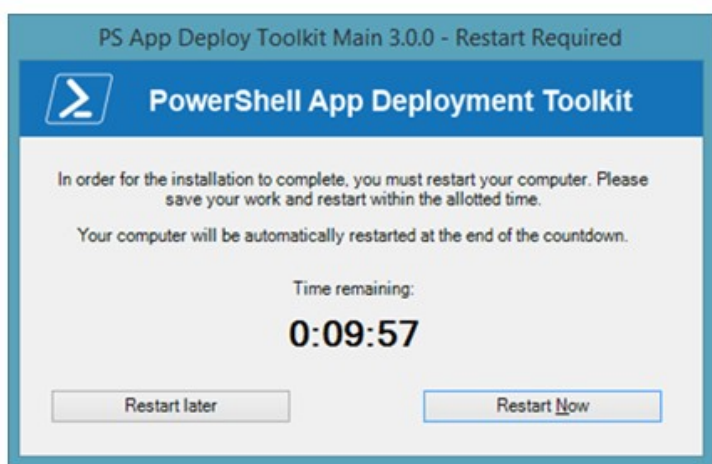
Tutkimuksen pohjalta havaittujen tarpeiden perusteella luotiin provisiointisovellus, jolla voidaan korvata kohdassa 4.2 mainittu puuttuva perinteinen esiasennuksen mahdollisuus. Tämän ratkaisun avulla madalletaan siirtymäkynnystä pilvipohjaisen työasemahallinnan käyttöönottoon kehityksen tässä vaiheessa. On selvää, että lähitulevaisuudessa Microsoft tulee lisäämään tarpeiden mukaisia ominaisuuksia Intune-ympäristöön ja todennäköisesti nyt kehitetty ratkaisu on pitkällä aikavälillä väliaikainen.

4.3.1 Sovelluksen kehitysalusta

Provisiointisovelluksen perustana käytetään valmista, yhteistökehitetty The PowerShell App Deployment Toolkit -työkalua, joka sisältää valmiita funktioita, joita yleisesti tarvitaan sovellusten ja työasemien käyttöönottovaiheessa (Cunningham, Lillis, Mashwani, & Motazedian, 2017). Yhdeksi oleelliseksi toimitteeksi itsepalvelullisuutta ajatellen nousee mahdollisuus tuottaa käyttäjälle interaktiivisia ilmoituksia asennuksen eri vaiheissa (Kuva 25) sekä asennustoimenpiteiden valmistumisesta (Kuva 26).



Kuva 25: PS Appdeployment toolkit, asennuksen tilatieto (Cunningham et al., 2017).



Kuva 26: PS Appdeployment toolkit, esimerkki-ilmoitus valmistumisesta (Cunningham et al., 2017).

Provisiointisovellus on Powershell-komentotulkkiin (Microsoft Corporation, 2017c) perustuva avoimen lähdekoodin sovellus, minkä päälle rakennetaan kohdeympäristön vaatimusten mukaiset toiminnallisuudet. Näihin toiminnallisuuksiin kuuluvat sovellusten asennukset, tietoturva-asetukset, rekisteripohjaiset asetukset, tiedostojen kopiointi jne. Yleisesti kaikki mahdolliset toimenpiteet, jotka halutaan automatisoida ja ovat tehtävissä ohjelmallisesti.

Tärkeimpiä valmiiksi suunniteltuja teknisiä toimintoja, jotka muutoin jouduttaisiin suunnittelemaan alusta lähtien, ovat muun muassa (Cunningham et al., 2017):

- Sovellusasennuksen MSI-paketin asennus.
- Minkä tahansa suoritettavan prosessin käynnistäminen.
- Laajennettu lokien käyttö.
- Tiedostojen hallittu kopiointi ja poistaminen.
- Käynnistyskuvakkeiden hallinta käynnistys-valikossa, työpöydällä ja tehtäväpalkissa.
- INI-tiedostojen hallinta.
- Windowsin rekisteriarvojen hallinta.

4.3.2 Provisiointisovelluksen paketointi ja julkaisu Intunessa

Provisiointisovellus ja siihen liittyvät sovellusasennusten tarvittavat tiedostot paketoidaan yhteen MSI-asennustiedostoon, joka julkaistaan ja kohdennetaan valitulle ryhmälle. Käytännössä jakelu kohdistetaan käyttäjätason ryhmälle, koska ennen käyttöönottovaihetta ei etukäteen tiedetä laitteiden nimiä ja tällä hetkellä ei ole mahdollista luoda laitekohtaisia tilejä ennakkoon Azure Active Directoryyn. Sovellusasennusten teknisen asennus tapahtuu laitetasolla, mutta asennusten kohdistaminen voidaan tästäkin huolimatta kohdentaa käyttäjällä (käytännössä käyttäjäryhmälle, jonka jäsenenä on käyttäjätunnus), jolloin mainostettu sovellus saadaan viiveellä sille työasemalle, mille käyttäjä on kirjautunut sisään. Tämän provisiointisovelluksen ja sen ympärille luodun käyttöönottoprosessin automaattisuus ja joustavuus perustuvat juuri tähän ajatukseen.

Sovellus julkaistaan Intune-hallinnassa Line of Business -sovellustyyppinä, joka käsittää myös MSI-paketoidut sovellukset. Edelleen sovellus kohdistetaan valitulle käyttäjäryhmälle (Kuva 27), jonka jäseniksi liitetään ne käyttäjätunnukset, joiden halutaan ottavan uuden työaseman itsepalvelullisesti käyttöön.

GROUP	TYPE
Enroll	Required ..

Kuva 27: Esimerkki provisiointisovelluksen jakelun kohdistamisesta (Savolainen, 2017b).

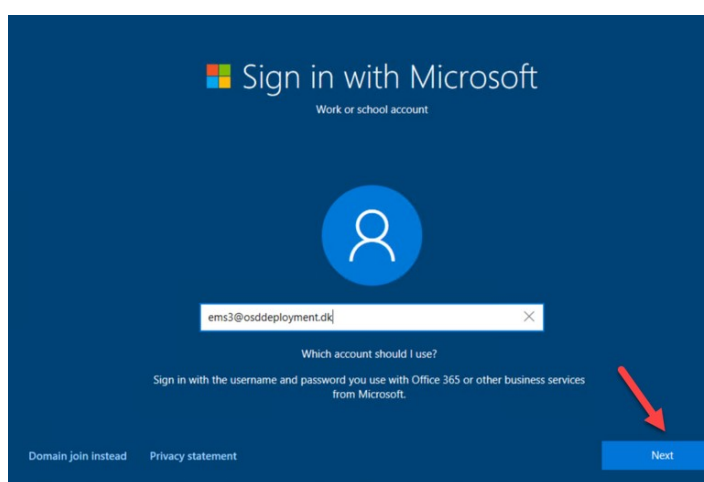
4.3.3 Suunniteltu työaseman käyttöönottoprosessi

Provisiointisovelluksen suunnittelun lähtökohtana on se, että uusia työasemia (usein miten kannettavia tietokoneita) ei enää perinteisesti esiasenneta tai yleisesti käytössä olevalla termillä ilmaistuna ei tehdä "image-asennusta". Käytännössä hyödynnetään laitteisiin räätälöityä laitevalmistajan tuottamaa OEM (Original Equipment Manufacturer)-asennusta, jolloin ensimmäisellä käynnistyksellä käyttäjä tyypillisesti valitsee muun muassa kieliasetukset ja aikavyöhykkeen sekä muita mahdollisia laitevalmistajan määrittämiä vaihtoehtoja.

Oleellisena osana itsepalvelullista käyttöönottoa (Kuva 28 Kuva 28) on ensimmäisellä käynnistyskerralla Azure Active Directory -hakemistopalveluun liittäminen (Kuva 29). Edelleen integraation kautta työasemalaite liittyy automaattisesti Intune-hallinnan piiriin ja lopulta ennalta julkaistu provisiointisovellus asentuu uuteen työasemalaitteeseen.

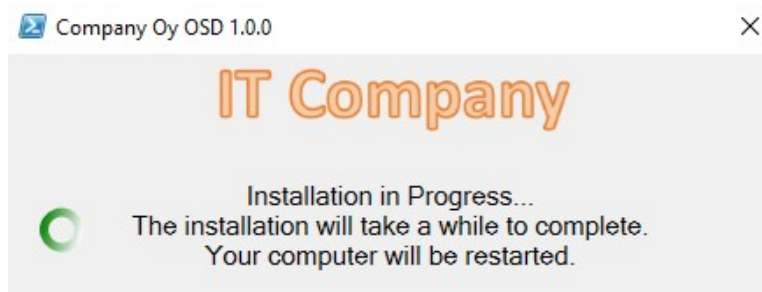


Kuva 28: Periaatteellinen käyttöönottoprosessi (Savolainen, 2017a)



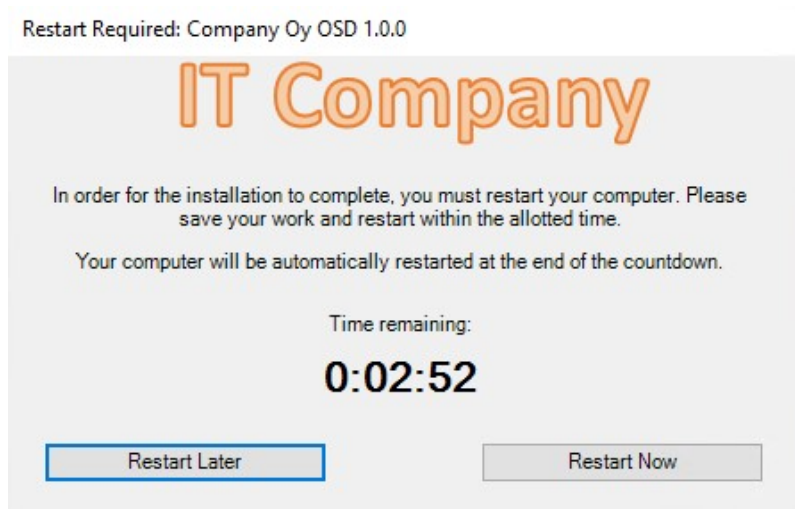
Kuva 29: Azure Active Directoryyn liittäminen (Larsen, 2017).

Azure AD -liitoksen jälkeen Windows 10 -käyttöjärjestelmän asennusohjelma suorittaa käyttöönoton loppuun ja kirjautuu työasemalla sisään käyttäjätunnuksella, jolla liitos on tehty. Pienellä viiveellä Azure-hallinnassa määritellyn integraation kautta työasema liittyy Intune MDM -hallinnan piiriin, minkä seurauksena ylläpitäjän asettama provisiointisovellus asennetaan ja kyseiselle työasemalle. Hallintaan liitoksen ja provisiointipaketin asentumisen aloitusvaiheiden välinen odotusaika riippuu käytössä olevasta internet-yhteyden tehollisesta nopeudesta sekä provisiointipaketin tiedostokoosta. Tiedostokoko puolestaan vaihtelee sen mukaisesti, kuinka paljon kohdeyrityksessä on haluttu tuoda alkuvaiheessa asennettavia liiketoimintasuoritusominaisuuksia mukaan käyttöönottovaiheeseen. Sisään kirjautuneen käyttäjätunnuksen työpöydällä on näkyvillä koko käyttöönottoprosessin ajan ilmoitus käynnissä olevasta asennuksesta (Kuva 30).



Kuva 30: Ilmoitus käynnissä olevasta asennusprosessista (Savolainen, 2017a).

Asennustoimenpiteiden valmistuttua sisään kirjautuneelle käyttäjälle tulee selväkielinen ilmoitus valmistumisesta ja kehote käynnistää työasema uudestaan (Kuva 31).



Kuva 31: Ilmoitus käyttöönottoprosessin valmistumisesta (Savolainen, 2017a).

5 Keskustelu ja arviointi

Microsoftin infraan liittyvästä aihepiiristä ei ole löydy kovin paljon tehtyjä tutkimuksia ylemmällä korkeakoulutasolla, kun alemmalla korkeakoulutasolla löytyy paljonkin erilaisten järjestelmien, muun muassa Configuration Managerin, käyttöönotoista ja hallinnasta. Erityisesti tämän opinnäytetyön näkökulmasta laadittuja, työasemahallinta suoraan pilvestä, tutkimuksia ei löydy julkisesta Internetistä lainkaan. Tämä heijastuu myös käytännön työelämään siten, että asiakasyritykset kertovat poikkeuksetta saavansa aihepiiristä paljon pintapuolisempaa tietoa muilta konsulttiyrityksiltä verrattuna opinnäytetyön tekijän edustamaan yhtiöön.

Aihepiiri on kuitenkin varsin kuuma ja ajankohtainen, varsinkin nyt alkusyksystä 2017 asiakasyritykset ovat heränneet päätelaitehallinnan siirtämisestä pilveen osittain tai kokonaan ja näin olleen on erittäin tärkeää pystyä tarjoamaan hyvää käytännön osaamista ja kokemusta pilvipohjaisen työasemahallinnan hyvistä puolista ja käytännön haasteista liittyen esimerkiksi tietoliikennekaistan riittävyys.

Ammattikorkeakoulussa edellytetään opinnäytetyön laadinnasta työelämälähtöisyyttä ja sen tulisi tuottaa todellista hyötyä jonkin osa-alueen, kuten tuotteistuksen, kehittämiseen ja sitä kautta myös liiketoiminnalliseen hyötyyn. Lisäksi opinnäytteen tekemisessä tulee käyttää runkona jotakin tutkimusmetodologiaa, jota noudattaen loogisesti analysoidaan kerättyä tutkimusaineistoa ja johdetaan lopulta tutkimustulokset ja vastaukset tutkimuskysymyksiin.

Tämän opinnäytetyön tavoitteina oli tapaustutkimuksena vertailla kahta Microsoftin hallintatuotetta, joista Configuration Manager on sisäverkossa käytettävä asiakaskohtainen tuote, ja toisaalta Microsoft Intune, joka on multitenantisti pilvipalveluna tarjottava tuote. Tutkimuksen tuloksena luotiin vertailutaulukoita, joiden avulla voidaan arvioida, mitä ominaisuuksia tullaan tarvitsemaan kohdeympäristössä ja millä tärkeysasteella. Tältä pohjalta asiakkuuskohdaisesti voidaan arvioida, ollaanko valmiita siirtymään pilvipohjaiseen hallintaan.

Tapaustutkimuksen tulosten perusteella arvioitiin suurimmaksi pilvihallitun työaseman haasteeksi esiasennusominaisuuden puuttuminen. Tähän ongelmaan kehitettiin suunnittelutieteellisenä tutkimuksena provisiointisovellus, joka näyttää käyttäjälle asennuksen vaiheet ja antaa lopuksi selkeän informaation lopullisesta valmistumisesta. Tällä ratkaisulla saavutettiin mahdollisuus realistisesti siirtää palvelukokonaisuus pilveen, mistä asiakkaat ovat olleet kiinnostuneita tämän tutkimuksen aikana. Kahdelle asiakkaalle ratkaisu on toteutettu tuotantoympäristöön.

5.1 Tapaustutkimuksen arviointi

Tutkimuksen reliabiliteettia arvioimalla vastataan kysymykseen, voidaanko kyseisellä tutkimuksella saadut tutkimustulokset toistaa (Hirsjärvi et al., 2009). Tässä kyseisessä tutkimuksessa vastakkain asettuvat pilvipalvelut ja sisäverkon eli ns. on premise -palvelut. Tutkimusdataa vertailtavista ominaisuuksia ja tarpeista kerättiin asiakasyritysten niiltä henkilöiltä, jotka käytännössä työskentelevät päivittäin näiden asioiden kanssa. Näin ollen näkökulma on varsin tekninen ja käytännöllinen, mutta ei kaupallinen. Kaupallinen näkökulma tällä hetkellä on yleisesti pilvipalveluiden puolella, koska se on epäilemättä kustannustehokkain vaihtoehto. Tällä perusteella oma arvioni on se, että reliabiliteetti on varsin hyvä, kun tutkimuksen kohderyhmän tyyppi pysyy samana.

Tutkimuksen validiteettia (Hirsjärvi et al., 2009) arvioimalla vastataan kysymykseen, vastaako tutkimustulos tutkimuskysymykseen (Hirsjärvi et al., 2009). Tapaustutkimuksen kysymykseksi oli asetettu: ”Millaisia työasemahallintatuotteen hallintaominaisuuksia voidaan tarkastella ja vertailla päätöksenteon pohjana valittaessa sisäverkkoon sijoitetun ja pilvestä hankitun palvelun välillä?”.

Tutkimusdataa kerättiin avoimella haastattelulla ja voidaan todeta, että haastattelua varten olisi ollut järkevää lisätä ominaisuuksien painotusasteikko. Nyt ominaisuuksien painotus tapahtui havainnoimalla, mitä pidän kuitenkin lopputuloksen kannalta hyvänä.

Tutkimuksessa selvitettiin tapaustutkimuksena Microsoft Intune- ja Microsoft System Center Configuration Manager -hallintatuotteiden ominaisuuksia ja niistä muodostettiin lopulta vertailutaulukoita sillä periaatteella, löytyykö kyseinen ominaisuus hallintatuotteesta vai ei. Edelleen vertailtavat ominaisuudet luokiteltiin seitsemään ryhmään, joista kukin muodostaa oman taulukon ominaisuuksien ryhmittelyn perusteella.

Vertailussa ei siis otettu kantaa siihen, voidaanko jokin ominaisuus saada käyttöön muilla keinoin kyseistä hallintatuotetta käytettäessä tai tarkemmin siihen, onko kyseinen ominaisuus yhtä hyvä kuin toisessa tuotteessa. Käytännössä tilanne onkin se, että pilviratkaisusta tarjottuna ominaisuus on lähes aina yksinkertaisempi ja ei-räätälöitävä verrattuna sisäverkon tuotteeseen, joka on lähes aina asiakaskohtainen ja räätälöitävissä halutulla tavalla. Räätälöintimahdollisuuksien vaikeustason arviointi voisi myös olla paikallaan, kun arvioidaan millä mallilla yleisesti halutaan lähteä liikkeelle.

Tapaustutkimusta aloitettaessa tutkijalla oli jo ennakonäkemyks suurimmasta kynnyskysymyksestä, eli esiasennusmahdollisuuden puutteesta, pilvipalveluihin siirtymisessä ja tätä varten

olikin jo aloitettu pilvipalvelussa hyödynnettävän provisiointityökalun kehittämisestä. Tutkimusdataa kerätessä käytännön työssä, tämä ajatus vahvistui lopullisesti.

5.2 Suunnittelutieteellisen tutkimuksen arviointi

Suunnittelutieteellisen tutkimuksen tavoitteena on tuottaa artefakti, joka tuo jotain uutta tai parantaa oleellisesti olemassa olevaa ratkaisua. Lisäksi artefaktin tulisi tutkimusongelman ratkaisun lisäksi tuottaa uutta tutkimustietoa aihealueella (Hevner & Chatterjee, 2010). Tässä valossa tutkijan on ymmärrettävä kohdeympäristö hyvin, jotta todelliset kehityskohteet ja ratkaistavat haasteet saadaan selville. Suunnittelututkimuksen DSRP-mallin arviointivaiheessa otetaan kantaa siihen, miten hyvin artefakti ratkaisee alkuperäisen tutkimusongelman.

Tämän suunnittelututkimuksen tutkimuskysymyksenä oli: ”Miten voidaan madaltaa kynnystä tarjota itsepalvelullinen työaseman käyttöönotto pilvipalvelusta tuotettuna?”. Taustalla on tutkijan pitkä kokemus päätelaitehallinnasta ja nykyinen paine siirtyä pilvipalveluiden käyttöön on tuonut esille erilaisia käytännön ongelmia yritysten valmiuden suhteen tällaiseen muutokseen. Tapaustutkimuksessakin esille tullut esiasennustoiminteen puute laitto liikkeelle ratkaisun kehitystyön, ja lopputuloksena syntynyt artefakti eli provisiointipaketti on nyt otettu käyttöön kahden asiakkuuden (käyttäjämäärät 500 ja 3000 käyttäjää) liiketoimintaympäristössä. Saadun palautteen perusteella käyttäjät pystyvät itsepalvelullisesti tekemään käyttöönoton ja erityistä on se, että uusi malli ei ole aiheuttanut merkittävässä määrin yhteydenottoja helpdesk-palveluihin.

Suunnittelutieteellisen tutkimuksen tavoitteena on tuottaa artefakti, jonka tulisi ratkaista alun perin määritelty tutkimusongelma ja toisaalta tuottaa merkittävää uutta tutkimustietoa aihepiiristä (Hevner et al., 2004). Lisäksi Marchin (et. al (1995) mukaan artefaktin on oltava uusi tai merkittävästi parempi kuin aiemmin julkaista vastaava tuotos. Tässä tutkimuksessa artefakti on provisiointisovellus, jonka tyypeistä ei ole saatavilla julkisesti tai kaupallisesti. Mahdollisesti joku on rakentanut vastaavan toiminnallisuuden, mutta ainakaan Suomen markkinoilta tällaista tietoa ei ole tullut julkisuuteen. Lisäksi tämän ratkaisun on todettu käytännössä madaltaneen kynnystä pilveen siirtymiseen työasemahallinnan osalta merkittävästi yhtiön asiakkuuksien ympäristöissä, joten voitaneen sano provisiointisovelluksen tuoneen merkittävää liiketoiminnallista hyötyä tutkimuksen tekijän työnantajalle.

Uutta tietoa tutkimus ei mielestäni tuonut tieteellisellä tasolla, joskin tutkimuksen aikana valmiiseen sovelluspohjaan kehitettiin lisätoimintoja, jotka ovat hyödyllisiä geneerisesti kaikissa ympäristöissä, eikä pelkästään yksittäisessä liiketoimintaympäristössä. Toki tieteelliseen tutkimukseen kuuluu myös tulosten jakaminen, mitä ei tehty tutkimuksen yhteydessä, mutta

laajasti katsoen kyseinen provisiointisovellus kehittyy jokaisen asiakkuuden myötä ja näin aina seuraavat asiakkaat hyötyvät edellisten asiakkuuksien myötä kehitetyistä toiminnallisuuksista.

References

- Arvidmark, J. (2017). Deployment research - giving back to the community. Retrieved from <http://deploymentresearch.com/>
- Badger, L., Grance, T., Patt-Corner, R., & Voas, J. (2012). *Cloud computing synopsis and recommendations*. (No. Special Publication 800-146). Gaithersburg: NIST(National Institute of Standards and Technology).
- Bamberg, J., Jokinen, P., & Laine, M. (2007). *Tapaustutkimuksen taito*. Helsinki: Gaudeamus.
- Baun, C., Kunze, M., Nimis, J., & Tai, S. (2011). *Cloud computing* (2nd edition ed.) Springer.
- Bennett, B., Daalmans, P., & Martinez, S. (2016). *Mastering system Center 2016 configuration manager* Sybex.
- Betts, D., Homer, A., Jezierski, A., Narumoto, M., & Zhang, H. (2012). *Developing multi-tenant applications for the cloud*. (3rd edition ed.) Microsoft Inc.
- Bojanova, I., Fiaidhi, J., Zhang, J., & Zhang, L. (2012). Enforcing multitenancy for cloud computing environments. *IT Professional*, (January), 10.4.2015.
- Chong, F., & Carraro, G. (2006). MSDN: Architecture strategies for catching the long tail. Retrieved from <https://msdn.microsoft.com/en-us/library/aa479069.aspx>
- Chong, F., Carraro, G. & Wolter, R. (2006). MSDN: Multi-tenant data architecture. Retrieved from <https://msdn.microsoft.com/en-us/library/aa479086.aspx>
- Citrix Techtarget. (2012). MDM and beyond: Rethinking mobile security in a BYOD world. Retrieved from http://docs.media.bitpipe.com/io_10x/io_108841/item_658490/030713_CTX_TT_WP_MDM-f-LO.pdf

- Coupland, M. (2014). *Microsoft system center configuration manager advanced deployment: Design, implement, and configure system center configuration manager 2012 R2 with the help of real-world example*. Birmingham, United Kingdom: Packt Publishing Ltd.
- Cunningham, D., Lillis, S., Mashwani, M. & Motazedian, A. (2017). PowerShell app deployment toolkit. Retrieved from <http://psappdeploytoolkit.com>
- Daahlmans, P. (2017). PeterDaahlmans.com. Retrieved from <https://configmgrblog.com>
- Gartner, I. Gartner IT glossary. Retrieved from <http://www.gartner.com/it-glossary>
- Gartner, I. (2006). Best practices in PC life cycle services. Retrieved from <https://www.gartner.com/doc/490086/best-practices-pc-life-cycle>
- Grance, T., & Mell, P. (2011). *The NIST definition of cloud computing*. (Special Publication No. 800-145). Gaithersburg: NIST (National Institute of Standards and Technology).
- Hevner, A., & Chatterjee, S. (Eds.). (2010). *Design research in information systems: Theory and practice*. New York, USA: Springer.
- Hevner, A., March, S., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1)
- Hirsjärvi, S., Remes, P., & Sajavaara, P. (2009). *Tutki ja kirjoita*. (15th ed.). Helsinki: Tammi.
- ISO. (2014). *Information technology – cloud computing – overview and vocabulary*. (International Standard No. ISO/IEC 17788:2014(E)). Switzerland: International Organization for Standardization.
- Järvinen, P., & Järvinen, A. (2004). *Tutkimustyön metodeista*. Tampere: Opinpajan kirja.
- Johnson, M. (2011). *Mobile device management: What you need to know for it operations management* Emereo Publishing.

- Kananen, J. (2013). *Case-tutkimus opinnäytetyönä*. Jyväskylä: Suomen Yliopistopaino Oy.
- Kansallinen Terveysarkisto (Kanta). (2015). Retrieved from <http://www.kanta.fi>
- Kaskisaari, I. (2016). In Onrego Sales (Ed.), *Email message: Intune hybrid mode vs. intune standalone*. Espoo: Microsoft Corporation.
- Kavis, M. (2014). *Architecting the cloud: Design decisions for cloud computing service models(SaaS, PaaS AND IaaS)*. Hoboken: John Wiley & Sons, Inc.
- Krsteovski, V. (2014). *Mastering system center configuration manager*. Birgmingham: Packt Publishing.
- Larsen, P. (2017). Mobile-first cloud-first. Retrieved from <https://osddeployment.dk>
- March, S., & Smith, G. (1995). Design and natural science research on information technology. *Decision Support Systems*, 15(4)
- Microsoft Corporation. (2017a). Enterprise mobility and security blog. Retrieved from <https://blogs.technet.microsoft.com/enterprisemobility/?product=microsoft-intune>
- Microsoft Corporation. (2017b). Microsoft intune documentation. Retrieved from <https://docs.microsoft.com/en-us/intune/>
- Microsoft Corporation. (2017c). Microsoft powershell. Retrieved from <https://msdn.microsoft.com/en-us/powershell/mt173057.aspx>
- Microsoft Corporation. (2017d). Overview of windows as service. Retrieved from <https://docs.microsoft.com/en-us/windows/deployment/update/waas-overview>
- Microsoft Corporation. (2017e). System center configuration manager. Retrieved from <https://www.microsoft.com/en-us/cloud-platform/system-center-configuration-manager>

- Microsoft Corporation. (2017f). What's a universal windows platform (UWP) app? Retrieved from <https://docs.microsoft.com/en-us/windows/uwp/get-started/whats-a-uwp>
- Microsoft Corporation. (2017g). What's new in version 1702 of system center configuration manager. Retrieved from <https://docs.microsoft.com/en-us/sccm/core/plan-design/changes/whats-new-in-version-1702>
- Nunamaker, J., & Briggs, R. (2011). Toward a broader vision for information systems. *ACM Transactions on Management Information Systems*, 2(4)
- Open Mobile Alliance. (2017). OMA specifications. Retrieved from <http://openmobilealliance.org/wp/index.html>
- Parker, A. (2016). Intune management choices. Retrieved from <http://stealthpuppy.com/windows-10-management-intune/>
- Peppers, K., Tuunanen, T., Rothenberger, M., & Chatterjee, S. (2008). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 24(3)
- Pierer, M. (2016). *Mobile device management: Mobility evaluation in small and medium-sized enterprises*. Vienna, Austria: Springer Vieweg.
- Robson, C. (2002). *Real world research*. Malden: Blackwell Publishing.
- Rymer, J. R., & Staten, J. (2012). *Understanding cloud's multitenancy*. (). Cambridge: Forrester Research, Inc.
- Saad-Sulonen, J. (2014). *Combining participations: Expanding the locus of participatory E-planning by combining participatory approaches in the design of digital technology and in urban planning*. (Doctoral thesis).
- Salo, I. (2010). *Cloud computing: Palvelut verkossa*. Porvoo: Docendo Oy.

Savolainen, S. (2017a). *Webinaari: Työasemat kuntoon itsepalvelun avulla. tietoisu toimitaan windows 10 työasemahallintaan.* (1st ed.). Espoo: Onrego Oy.

Savolainen, S. (2017b). *Microsoft intune -hallintaympäristö(kehitysympäristö).*

Savolainen, S. (2017c). *Microsoft SCCM -hallintaympäristö(kehitysympäristö).*

Sosinsky, B. (2011). *Cloud computing bible.* Indianapolis: Wiley Publishing, Inc.

van Aken, J. E., & Romme, G. L. (2004). *A design science approach to evidence-based management.*

Waschke, M. (2012). *Cloud standards: Agreements that hold together clouds* Apress.

Yin, R. K. (2014). *Case study research: Design and methods.* (5. Edition ed.,). USA: Sage Publications, Inc.

Kuvat

Kuva 1: Yinin (2014) lineaaris-iteratiivinen prosessi.....	9
Kuva 2: Suunnittelututkimuksen prosessimalli Peffersin (2008) mukaan.	14
Kuva 3: Pilvipalvelumallit (Baun et al., 2011).	23
Kuva 4: Cloud Stack (Kavis, 2014).	24
Kuva 5: Pilvilaskennan yleiskuva (Kavis, 2014).	26
Kuva 6: Yksityisen pilven toimijamalli (Waschke, 2012).	27
Kuva 7: Julkisen pilven toimijamalli (Waschke, 2012).	28
Kuva 8: Hybridimalli (Badger et al., 2012).	29
Kuva 9: Eristäminen vs. jaettu ympäristö.	35
Kuva 10: Tietokannan rakenne multitenantissa ympäristössä (Chong et al., 2006).....	37
Kuva 11: Erilliset tietokannat (Chong et al., 2006).	38
Kuva 12: Jaettu tietokanta, erilliset skeemat (Chong et al., 2006).	38
Kuva 13: Jaettu tietokanta, jaettu skeema (Chong et al., 2006).	39
Kuva 14: Elinkaarimalli.	41
Kuva 15: SCCM:n esiasennusprosessi (Savolainen, 2017c).	43
Kuva 16: Java 8 -sovelluksen asennus- ja poistokomennot (Savolainen, 2017c).....	44
Kuva 17: SCCM:n sovellusjakelutyytit (Savolainen, 2017c).	44
Kuva 18: Esimerkki velhopohjaisesta asetuksesta (Savolainen, 2017c).	46
Kuva 19: Esimerkki räätälöitävästä konfiguraatiosta (Savolainen, 2017c).	46
Kuva 20: SCCM:llä jaeltavat profiilityytit yrityksen toimintaympäristöön pääsyä varten (Savolainen, 2017c).....	47
Kuva 21: EndPoint Protection -sovelluksen tukemat etähallintakomennot (Savolainen, 2017c).	47
Kuva 22: SCCM:n etätukiyhteys (Savolainen, 2017c).	48
Kuva 23: Sähköpostiprofiilin asettaminen Intune-hallinnassa (Savolainen, 2017b).	49
Kuva 24: Työasemalaitteita vanhasta Intunen hallintanäkymässä (Savolainen, 2017b).	50
Kuva 25: PS Appdeployment toolkit, asennuksen tilatieto (Cunningham et al., 2017).....	55
Kuva 26: PS Appdeployment toolkit, esimerkki-ilmoitus valmistumisesta (Cunningham et al., 2017).	55
Kuva 27: Esimerkki provisiointisovelluksen jakelun kohdistamisesta (Savolainen, 2017b).	56
Kuva 28: Periaatteellinen käyttöönottoprosessi (Savolainen, 2017a)	57
Kuva 29: Azure Active Directoryyn liittäminen (Larsen, 2017).	57
Kuva 30: Ilmoitus käynnissä olevasta asennusprosessista (Savolainen, 2017a).	58
Kuva 31: Ilmoitus käyttöönottoprosessin valmistumisesta (Savolainen, 2017a).	58

Taulukot

Taulukko 1: Tutkimusattribuutit	19
Taulukko 2: Multitenanttisuuden toteutus pilvipalvelutasoilla.	31
Taulukko 3: Yleiset ominaisuudet.....	51
Taulukko 4: Esiasennusominaisuudet.....	52
Taulukko 5: Sovellushallintaominaisuudet.	52
Taulukko 6: Päivitysten hallintaominaisuudet.	52
Taulukko 7: Raportointiominaisuudet.	52
Taulukko 9: Asetusten hallinta (engl. configuration policy).	52
Taulukko 10: Etähallinta- ja etätukitoimet (engl. remote tasks ja remote assistance).....	53